4142. 70026
312.360.0080

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２００４年 ２月 ２日

出 願 番 号
Application Number: 特願２００４－０２５６７７

[ST. 10/C]: [ＪＰ２００４－０２５６７７]

出 願 人
Applicant(s): 富士通株式会社

２００４年 ２月１６日

特許庁長官
Commissioner,
Japan Patent Office 今 井 康 夫

出証番号 出証特２００４－３００９４９７

【書類名】　特許願
【整理番号】　0352336
【提出日】　平成16年 2月 2日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 19/00
【発明者】
　【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
　【氏名】　土河原　信雄
【特許出願人】
　【識別番号】　000005223
　【氏名又は名称】　富士通株式会社
【代理人】
　【識別番号】　100103528
　【弁理士】
　【氏名又は名称】　原田　一男
【先の出願に基づく優先権主張】
　【出願番号】　特願2003-283314
　【出願日】　平成15年 7月31日
【手数料の表示】
　【予納台帳番号】　076762
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9909129

【書類名】特許請求の範囲
【請求項１】
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスが未登録であるか判断するステップと、
特定された前記業務クラスの項目が未登録である場合には、前記業務クラス管理部に当該特定された前記業務クラスの項目を登録するステップと、
を含み、コンピュータにより実行される情報処理方法。
【請求項２】
特定された前記業務クラスの項目が未登録である場合には、当該特定された前記業務クラスのための雛型ソースプログラム・データを生成し、ソースプログラム格納部に格納するステップ
をさらに含む請求項１記載の情報処理方法。
【請求項３】
帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する業務クラス特定ステップと、
予め定義され且つメモリにロードされた業務クラスの中で、特定された前記業務クラスを呼び出す呼出ステップと、
を含み、コンピュータにより実行される情報処理方法。
【請求項４】
前記帳票画面に係る帳票に対応して当該帳票全体の処理シーケンスが規定されておらず、呼び出された前記業務クラスの処理にて当該帳票に対する処理が完了するように、前記業務クラスの各々が構成されていることを特徴とする請求項３記載の情報処理方法。
【請求項５】
帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する業務クラス特定ステップと、
予め定義され且つメモリにロードされた業務クラスの中で、特定された前記業務クラスを呼び出す呼出ステップと、
をコンピュータに実行させるプログラム。

【書類名】明細書
【発明の名称】ＸＭＬドリブンアーキテクチャにおける情報処理方法及びプログラム
【技術分野】
【０００１】
本発明は、ＸＭＬ（eXtensible Markup Language）をベースとしたプログラム開発技術及びＸＭＬをベースとした情報処理技術に関する。
【背景技術】
【０００２】
従来では、画面及び入出力データ、プログラムの機能分担、プログラムの構造、データベースのデータ項目などを一元的な観点で考慮した上で、業務プログラムの開発がなされていた。このため、１つの業務プログラムの同時並列的な設計は不可能であり、業務プログラムの規模が大きくなると開発期間が長期化することも多かった。また、仕様の変更・追加を行う際には、全面的な見直しが必要となり、検討時間の増加、及び修正時間の増加を生じていた。
【０００３】
このような問題を解決すべく特開平９－１９８２４０号公報は、画面の設計時に画面上の入力フィールドに入力される文字数や文字の入出力定義体を生成し、さらにモックアップに用いられていたプログラムエントリの工程をそのままプログラムの製造工程に自動継承させて、各業務ルーチンの設計を容易に行えるようにする技術を開示している。
【特許文献１】特開平９－１９８２４０号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
しかしながら、特許文献１に開示された技術では、入出力定義体の自動生成やプログラムエントリの工程の自動継承は容易になったが、自動継承された設計情報を基に業務個別ルーチンを作成しなければならず、プログラムを同時並行して作成する場合、プログラムの作成漏れなどが生じてしまうことがある。また、プログラム実行の際に行われる入力データの受け渡しは、設計時に決定したＩＤにより行われているので、後に修正などする場合には、その設計者以外には分かりにくい構成となっている。
【０００５】
従って、本発明の目的は、より開発期間を短縮でき且つより品質の高い業務プログラムの開発を可能にするための技術を提供することである。
【０００６】
また本発明の他の目的は、本発明に係るプログラム開発技術により作成された業務プログラムを実行するための基盤技術を提供することである。
【課題を解決するための手段】
【０００７】
本発明の第１の態様に係る情報処理方法は、ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された業務クラスの項目が未登録であるか判断するステップと、特定された業務クラスの項目が未登録である場合には、業務クラス管理部に当該特定された業務クラスの項目を登録するステップとを含む。
【０００８】
帳票画面に対応するＸＭＬデータのタグをベースに業務クラスを作成する際に、業務クラス管理部において、生成すべき業務クラスの項目を管理するため、重複する業務クラスを作成したり、不足が生じたりすることなくプログラム開発を行うことができる。なお、ＸＭＬのタグに対応する業務クラスを作成すればよいので、全体を把握せずとも業務クラスを作成でき、業務プログラムの品質向上が容易で、開発期間の短縮も図ることができる。

【０００９】
また、特定された業務クラスの項目が未登録である場合には、当該特定された業務クラスのための雛型ソースプログラム・データ（例えばファイル）を生成し、ソースプログラム格納部に格納するステップをさらに含むようにしてもよい。プログラム作成者は、生成された雛型ソースプログラム・データを用いて所定の機能を実現するためのプログラムを作成すればよく、過不足無く業務クラスを用意できる。

【００１０】
さらに、帳票項目についてのデータを格納するためのオブジェクトの雛型ソースファイルについても、業務クラスと同様な処理にて生成するようにしても良い。

【００１１】
本発明の第２の態様に係る情報処理方法は、例えば上記のような方法を用いて作成された業務クラスの実行方式に関する情報処理方法であって、帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する業務クラス特定ステップと、予め定義され且つメモリにロードされた業務クラスの中で、特定された業務クラスを呼び出す呼出ステップとを含む。

【００１２】
業務クラスは元々ＸＭＬデータに含まれるタグに対応して作成されているため、特定の選択又は入力データを含むＸＭＬデータが入力された場合には、当該ＸＭＬデータに含まれるタグに対応する業務クラスを呼び出すことにより、特定の選択又は入力データを処理するものである。

【００１３】
また、帳票画面に係る帳票に対応して当該帳票全体の処理シーケンスが規定されておらず、呼び出された前記業務クラスの処理にて当該帳票に対する処理が完了するように、業務クラスの各々が構成されるようにしてもよい。このように、ＸＭＬデータのタグに対応するクラスにより必要な業務ロジックが実装されれば、システム開発の効率が向上する。
なお、タグに対応する業務クラスが存在しない場合には、不適切なＸＭＬデータであるから、エラーを通知するような構成であってもよい。

【００１４】
また、帳票画面への入力又は選択データを帳票画面を表示した装置から受信し、当該帳票画面への入力又は選択データに対応するタグを含むＸＭＬデータを生成するステップをさらに実行するようにしても良い。

【００１５】
さらに、帳票画面に対する入力又は選択データを、予め定義され且つメモリにロードされた帳票項目格納オブジェクトに出力し、当該帳票項目格納オブジェクトにより帳票画面に対する入力又は選択データをメモリに格納するステップと、呼び出された業務クラスと帳票項目格納オブジェクトとがデータをやり取りすることにより処理を実施する処理実施ステップとをさらに含むようにしても良い。

【００１６】
さらに、呼び出された業務クラスにより出力要求を受けた場合、帳票項目格納オブジェクトに保持されたデータを用いて出力用ＸＭＬデータを生成し、メモリに格納するステップと、メモリに格納された出力用ＸＭＬデータを帳票画面を表示した装置に出力するステップとをさらに実行するようにしても良い。

【００１７】
また、本発明に係る情報処理方法をコンピュータに実行させるためのプログラムを作成することも可能であって、当該プログラムは、例えばフレキシブル・ディスク、ＣＤ－ＲＯＭ、光磁気ディスク、半導体メモリ、ハードディスク等の記憶媒体又は記憶装置に格納される。プログラム等は、ネットワークを介してデジタル信号として配信される場合もある。また、処理途中のデータについては、コンピュータのメインメモリ等の記憶装置に一時保管される。

【発明の効果】
【００１８】
本発明によれば、より開発期間を短縮でき且つより品質の高い業務プログラムの開発を行うことができる。
【００１９】
また別の側面として、本発明に係るプログラム開発技術により作成された業務プログラムを実行するための基盤技術を提供できる。
【発明を実施するための最良の形態】
【００２０】
１．業務クラス（プログラム）の作成
図１に、業務クラスの作成時における、本発明の一実施の形態に係るコンピュータ・システムの機能ブロック図を示す。コンピュータ・システム１００は、開発中のシステムにおいて表示される画面の構成を行い、当該画面に対応するＨＴＭＬ（Hyper Text Markup Language）ファイルなどを生成する画面作成部１と、画面作成部１で生成されたＨＴＭＬファイルを格納するＨＴＭＬファイル格納部３と、ＨＴＭＬファイル格納部３に格納されたＨＴＭＬファイルを解析し、所定のルールに従ってＸＭＬデータに変換するＸＭＬ変換部５と、ＸＭＬ変換部５により生成されたＸＭＬデータ等を格納するＸＭＬデータ格納部７と、ＸＭＬデータ格納部７に格納されたＸＭＬデータを解析して、共通インターフェースファイル群１１の少なくとも一部及び雛型業務ソースファイル群１３を生成するインターフェースマネージャ９と、インターフェースマネージャ９により参照及び更新される業務クラス管理ファイル１５とを含む。
【００２１】
画面作成部１は、一般に販売されているホームページエディタなどであり、ユーザがＨＴＭＬのタグを記述せずに画面構成を行うとＨＴＭＬファイルなどを生成するものである。例えば図１に示すように、銀行名、支店名、口座種別、口座番号及び取引金額の入力欄を含む入金取引の画面１ａを構成し、対応するＨＴＭＬファイル等が生成されるものとする。生成されたＨＴＭＬファイルはＨＴＭＬファイル格納部３に格納される。
【００２２】
ＸＭＬ変換部５は、一般に販売されているＸＭＬ　ＳＰＹ（ＩＣＯＮ社の商標）等のＸＭＬ文書作成支援ツールであり、ＨＴＭＬファイル格納部３に格納されたＨＴＭＬファイルを読み込み、ＸＭＬデータ、スタイルシート（ＸＳＬ（eXtensible Style Language）データ）、スキーマを生成し、ＸＭＬデータ格納部７に格納するものである。なお、以下では、本実施の形態のポイントの把握を容易にするため、ＸＭＬデータについての説明を中心に行っていくが、ＸＳＬデータには、ＸＭＬデータを画面表示するためのデータ（書式等）が含まれる。またスキーマには、ＸＭＬデータの妥当性検査を行うためのデータ（タグ構造等）が含まれる。図１に示すように、画面１ａのためのＨＴＭＬファイルから、入金取引入力というタグ（開始及び終了タグ。以下同じ。）と、銀行名というタグと、支店名というタグと、口座種別というタグと、口座番号というタグと、取引金額というタグとを含むＸＭＬデータ５ａが生成される。また併せて、ＸＭＬデータ５ａに対応するＸＳＬデータ及びスキーマが生成される。
【００２３】
雛型業務ソースファイル群１３には、インターフェースマネージャ９により生成され且つＸＭＬデータに含まれるタグに対応する雛型ソースプログラムのファイルが含まれる。図１に示すように、インターフェースマネージャ９がＸＭＬデータ５ａを処理すると、入金取引入力という伝票の前処理を行うための「入金取引入力前処理」という業務クラス１３１と、銀行名タグに対応する「銀行名」という業務クラス１３２と、支店名タグに対応する「支店名」という業務クラス１３３と、口座種別タグに対応する「口座種別」という業務クラス１３４と、口座番号タグに対応する「口座番号」という業務クラス１３５と、取引金額タグに対応する「取引金額」という業務クラス１３６と、入金取引入力という伝票処理に対応する「入金取引入力」という業務クラス１３７と、入金取引入力という伝票

の後処理を行うための「入金取引入力後処理」という業務クラス１３８とが、雛型ソースファイルとして生成される。雛型ソースファイルは、業務クラスとしての基本的なソースコードを含むだけであって、後で業務ロジックを例えばメソッドとしてコーディングする必要がある。なお、雛型ソースファイルは、空であってもよい。

【００２４】

以下に口座番号という業務クラスの雛型ソースファイルの一例を示す。なお、この具体的内容についてはＪａｖａ（Sun Microsystems社の登録商標）言語によって記述された一般的なものであるから、詳細な説明については省略する。

【００２５】

[表１]

```
package com.fff.xxxx.usr;

import com.fff.xxxx.common.*;
import com.fff.xxxx.util.*;
import com.fff.xxxx.processor.*;
//クラス：USR_口座番号
public class USR_口座番号 implements XxxxUserProcessLibrary
{
 //コンストラクタ
 public USR_口座番号()
 {
  //ここにコンストラクタでの処理内容を記述してください。
 }

 //プロセス実行関数
 public void process(XxxxInterface rinterface) throws Exception
 {
  //オブジェクトプールから自領域を取得する。
  String 口座番号 = (String)rinterface.getValue_XML("口座番号");

  //ここに処理単位毎のプログラムを記述してください。
 }
}
```

【００２６】

また、共通インターフェースファイル群１１には、伝票項目域１１１と、プロセッサ予約域１１２と、クラス間インターフェース領域１１３と、アプリケーション定数域１１５と、データベース（ＤＢ）情報域１１７と、ユーザがエディタ等を用いて作成する初期設定ファイル１１９とが含まれる。初期設定ファイル１１９を除くこれらの領域は、例えば複数の業務クラスにより共通に用いられるオブジェクトの雛型ソースプログラム（以下、共通インターフェース雛型ソースファイルと呼ぶ。）が格納される領域である。共通インターフェース雛型ソースファイルは、上で述べた業務クラスの雛型ソースファイル同様、基本的なソースコードを含むだけであって、後でロジックを例えばメソッドとしてコーディングする必要がある。但し、共通インターフェース雛型ソースファイルは、空であってもよい。

【００２７】

伝票項目域１１１には、伝票項目データを格納するためのオブジェクトの共通インターフェース雛型ソースファイルが格納される。伝票項目域１１１に格納される共通インターフェース雛型ソースファイルは、インターフェースマネージャ９によってＸＭＬデータに基づき生成される。例えば、ＸＭＬデータに含まれる伝票項目の各々に対応して共通インターフェース雛型ソースファイルが生成される。

【００２８】
以下に、口座番号という伝票項目に係る共通インターフェース雛型ソースファイルの一例を示す。但し、この具体的内容は、Ｊａｖａ言語によって記述された一般的なものであるから、詳細な説明については省略する。
【００２９】
［表２］

```
package com.fff.xxxx.pool;

import com.fff.xxxx.*;

import com.fff.xxxx.common.*;

import java.util.*;
////LOG出力用
/*LOG*/import com.fff.OPERAPARTS.com.SSCLG00C;

/** クラス：com_口座番号 */
public class com_口座番号 implements XxxxCommonInLibrary{


    /** ハッシュの何番目に格納するかのフラグ */
    public int flagGet = 0;
    /** ハッシュの何番目から取り出したかのフラグ */
    public int flagSet = 0;
    /** エラーハッシュの何番目に格納するかのフラグ */
    public int errflagGet = 0;
    /** エラーハッシュの何番目から取り出したかのフラグ */
    public int errflagSet = 0;
        /** 子タグクラスを格納するハッシュテーブル*/
        private Hashtable childtags = new Hashtable();

        /** 値を格納するハッシュテーブル*/
        private Hashtable values = new Hashtable();

        /** エラーコードを格納するハッシュテーブル*/
        private Hashtable errors = new Hashtable();

        /** 属性ハッシュを管理するハッシュテーブル*/
        private Hashtable attributes = new Hashtable();

        /** 属性値を格納するハッシュテーブル*/
////LOG出力用
/*LOG*/ private SSCLG00C ewkoSsclg00c;
    /** コンストラクタ */
    public com_口座番号(){

        //ここにコンストラクタでの処理内容を記述してください。

    ////LOG出力用
    /*LOG*/ ewkoSsclg00c = new SSCLG00C(this.getClass().getName());
```

```
    }

    /** プロセス実行関数 */
    public void process(XxxxInterface rinterface){

        ewkoSsclg00c.mvLogWrite(3,"プロセス呼び出しクラス名    : com_口座番号.pr
ocess()");
    }


        /** setメソッド */
        public void set(Object val){
            String numstr = String.valueOf(flagGet);
            values.put(numstr, val);
            flagGet += 1;
        }
        /** getメソッド */
        public Object get(int key){
            String numstr = String.valueOf(key);
            return values.get(numstr);
        }
        /** getメソッド最初の項目を取得 */
        public Object get(){
        String numstr = String.valueOf(flagGet - 1);
            return values.get(numstr);
        }
        /** valueテーブルの中身をクリアする*/
        public void clearValue(){
        attributes.clear();
        values.clear();
        clearStatus();
        flagGet = 0;
        }
        //エラーコードを格納する
        public void setStatus(int value){
        String numstr = String.valueOf(errflagGet);
        errors.put(numstr, new Integer(value));
        errflagGet += 1;
        }
        //エラーコードを取得する
        public int getStatus( String key){
            return ((Integer)errors.get(key)).intValue();
        }
        //エラーコードを取得する
        public int getStatus(){
            String numstr = String.valueOf(errflagGet - 1);
            return ((Integer)errors.get(numstr)).intValue();
        }
        /** エラーコード格納テーブルの中身をクリアする*/
```

```
    public void clearStatus(){
    errors.clear();
    errflagGet = 0;
    setStatus(0);
    }
    //繰り返し数を取得する
    public int getNumber(){
        return values.size();
    }
    //属性の値を取得
    public Object getAttr(int num, String attsname){
        String numstr = String.valueOf(num);
        Hashtable attshash = (Hashtable)attributes.get(attsname);
        return attshash.get(numstr);
    }
    //属性に値をセット
    public void setAttr(int num, String attsname, Object attsVal){
        String numstr = String.valueOf(num);
        Hashtable attshash = (Hashtable)attributes.get(attsname);
        attshash.put(numstr, attsVal);
    }
    //属性名を戻す
    public Enumeration getAttrNames(){
        return attributes.keys();
        }
    //属性名から、属性値の入ったHashを返す
    public Hashtable getAttrs(String attsname){
        return (Hashtable)attributes.get(attsname);
    }
}
```

【００３０】

プロセッサ予約域１１２には、業務クラスの実行処理に関わるプロセッサ（図６のプロセッサ２０７１）の制御に用いるデータを格納するためのオブジェクトの共通インターフェース雛型ソースファイルが格納される。プロセッサ予約域１１２に格納される共通インターフェース雛型ソースファイルは、インターフェースマネージャ９が有する画面入力機能（ユーザインターフェース）を介して設定されたデータ項目に応じて、インターフェースマネージャ９により生成され、プロセッサ予約域１１２に格納される。画面入力機能とは、ユーザからのデータ入力を受け付ける画面を表示させ、入力されたデータに基づき項目を設定する機能である。

【００３１】

クラス間インターフェース領域１１３には、業務クラス間のインターフェース項目に関するデータを格納するためのオブジェクトの共通インターフェース雛型ソースファイルが格納される。なお、業務クラス間のインターフェース項目とは、例えば複数の業務クラス間におけるデータの受け渡しを行うために用いる項目である。クラス間インターフェース領域１１３に格納される共通インターフェース雛型ソースファイルは、プロセッサ予約域１１２に格納される共通インターフェース雛型ソースファイル同様、インターフェースマネージャ９が有する画面入力機能を介して設定されたデータ項目に応じて、インターフェースマネージャ９により生成され、プロセッサ予約域１１２に格納される。

【００３２】

アプリケーション定数域１１５には、業務クラスが参照する定数項目データを格納する

ためのオブジェクトの共通インターフェース雛型ソースファイルが格納される。なお、業務クラスが参照する定数項目とは、初期設定ファイル１１９において規定されているデータに基づき設定される項目であり、例えばプログラム実行可能時間帯等の業務ルールに関する項目である。アプリケーション定数域１１５に格納される共通インターフェース雛型ソースファイルは、プロセッサ予約域１１２やクラス間インターフェース領域１１３に格納される共通インターフェース雛型ソースファイル同様、インターフェースマネージャ９が有する画面入力機能を介して設定されたデータ項目に応じて、インターフェースマネージャ９により生成され、アプリケーション定数域１１５に格納される。

【００３３】

ＤＢ情報域１１７には、ＤＢのテーブル定義項目データを格納するためのオブジェクトの共通インターフェース雛型ソースファイルが格納される。ＤＢ情報域１１７に格納される共通インターフェース雛型ソースファイルは、伝票項目域１１１に格納される共通インターフェース雛型ソースファイル同様、インターフェースマネージャ９によってＸＭＬデータに基づき生成される。例えば、ＸＭＬデータに含まれる伝票項目の各々に対応して共通インターフェース雛型ソースファイルが生成される。

【００３４】

初期設定ファイル１１９は、上で述べた定数項目データを含むファイルであり、例えばＸＭＬ形式のデータである。業務クラスの実行処理において、このファイルは、業務クラスによる処理を実行する際に読み込まれ、対応するオブジェクトに定数項目データが保持される。

【００３５】

業務クラス管理ファイル１５は、例えば図２に示すようなデータを含む。図２には、ＸＳＬファイルを指定していないＸＭＬデータ（業務クラス管理ファイル１５）をＸＭＬ対応のブラウザで表示した場合の画面例が示されている。図２の例では、資産管理タグにより、業務クラス管理ファイル１５により開発するプログラムにおいて必要とされる資産の管理を行っていることを表している。また、本実施の形態では業務クラスだけではなく、共通インターフェースファイル群１１についても管理するようになっており、オブジェクトプール・タグが設けられている。ここでオブジェクトプールとは、共通インターフェース雛型ソースファイルに基づき実行可能なクラスが生成されている場合に、処理実行時に当該クラスがロードされるメモリ領域を指す。オブジェクトプール・タグ配下には、プロセッサ予約域１１２に対応するプロセッサ予約域タグと、伝票項目域１１１に対応する伝票項目域タグと、ＤＢ情報域１１７に対応するＤＢ情報域タグと、アプリケーション定数域１１５に対応するアプリ定数項目域タグと、クラス間インターフェース領域１１３に対応するクラス間インターフェース項目域タグとが設けられている。図４の例では、「＋」印でプロセッサ予約域タグ及びクラス間インターフェース項目域タグの配下には既に定義された項目があることを示している。一方、ＤＢ項目域タグ及びアプリ定数項目域タグの配下にはまだ定義されている項目はない。ここで「項目が定義されている」とは、項目に対応する共通インターフェース雛型ソースファイルが生成され、共通インターフェースファイル群１１のいずれかの領域に格納されているということを指す。

【００３６】

いずれの領域に対応する共通インターフェース雛型ソースファイルについてもインターフェースマネージャ９により各領域に格納するようになっている。そして、インターフェースマネージャ９は共通インターフェース雛型ソースファイルをいずれかの領域に格納すると、当該領域に対応するタグの配下に、共通インターフェース雛型ソースファイルに対応する項目を登録する。

【００３７】

図２では、伝票項目域タグ配下には既に定義された項目が存在しており、それが具体的に示されている。図１の画面１ａに対応して、「入金取引入力」タグ、「取引金額」タグ、「口座番号」タグ、「口座種別」タグ、「支店名」タグ及び「銀行名」タグが設けられている。「入金取引入力」は伝票に対応するため「ルート判定」という属性が規定されて

いる。また各タグでは、登録日時という属性が規定されている。

【００３８】

また、図２の例では、生成すべき業務クラス（すなわち生成した雛型業務ソースファイル）が、雛型業務クラス・タグの配下に規定されている。ここでは、取引金額タグ、口座番号タグ、口座種別タグ、支店名タグ、銀行名タグ、入金取引入力前処理タグ、入金取引入力後処理タグ及び入金取引入力タグとが含まれる。なお各タグには、登録日時という属性が規定されている。またタグ名の前にＵＳＲとあるのは、ユーザにより用意されるべきクラスであることを示すものである。

【００３９】

次に図３を用いて図１に示したコンピュータ・システム１００の処理フローを説明する。まず本コンピュータ・システム１００のユーザは、現行の伝票、帳票、画面構成を検討すると共に、これから開発する新たなコンピュータ・システム用に追加又は変更すべき機能を把握し、画面作成部１により新たなコンピュータ・システム用の画面構成を行う。これにより、画面作成部１は、ＨＴＭＬファイルなどを生成し、ＨＴＭＬファイル格納部３に格納する（ステップＳ１）。なお、ツールなどを用いたりせずともＨＴＭＬファイルの作成は可能であるため、本ステップについては点線ブロックで示している。また、本ステップの処理を行うためのツールについては、インターフェースマネージャ９とは別にユーザが用意することも可能である。

【００４０】

次に、ＸＭＬ変換部５は、ＨＴＭＬファイル格納部３に格納されたＨＴＭＬファイルを読み出して、ＸＭＬデータを生成し、ＸＭＬデータ格納部７に格納する（ステップＳ３）。この際、ＸＭＬデータに対応するＸＳＬデータ及びスキーマを併せて生成し、ＸＭＬデータ格納部７に格納しておく。なお、本ステップの処理を行うためのツールについても、インターフェースマネージャ９とは別にユーザが用意することも可能であり、点線ブロックで表されている。次に、インターフェースマネージャ９は、プロセス初期処理及び終了処理クラスのための雛型ソースプログラムを生成し、例えば雛型ソースファイル群１３を格納する記憶装置に格納する（ステップＳ５）。プロセス初期処理及び終了処理クラスのためのソースプログラムについても雛型であり、プロセス初期処理及び終了処理のための具体的な機能を実現するためのプログラムは、後にコーディングされる。

【００４１】

そして、インターフェースマネージャ９は、ＸＭＬデータ格納部７を参照し、ＸＭＬデータの未処理タグを１つ選択する（ステップＳ７）。そして、業務クラス管理ファイル１５の雛型業務クラス・タグの配下を参照して、ステップＳ７で選択されたタグが登録されていないか確認する（ステップＳ９及びＳ１１）。業務クラス管理ファイル１５の雛型業務クラス・タグの配下に既にタグが登録されているということは、既に雛型業務ソースファイルも生成されているということであり、再度生成すると重複を生じてしまうために確認する。もし、既に登録されていると判断された場合には、ステップＳ２３に移行する。一方、未登録であることが確認できれば、当該タグが先頭タグであるか判断する（ステップＳ１３）。すなわち、ＸＭＬデータにおいて伝票を特定するタグであるか判断する。先頭タグであると判断された場合には、当該伝票の前処理、後処理及び伝票クラスの雛型ソースプログラムを生成し、例えば雛型ソースファイル群１３を格納する記憶装置に格納する（ステップＳ１５）。そしてステップＳ２１に移行する。一方、先頭タグではないと判断された場合には、当該タグが終了タグであるか判断する（ステップＳ１７）。例えば伝票を特定するタグの終了タグであるかを判断する。もし、終了タグであると判断された場合にはステップＳ２３に遷移する。一方、終了タグではないと判断された場合には、通常のタグであるので、当該タグに対応する業務クラスの雛型ソースプログラムを生成し、例えば雛型ソースファイル群１３を格納する記憶装置に格納する（ステップＳ１９）。上でも述べたが、雛型ソースプログラムは、基本的なソースコードのみが含まれるだけのものであり、それのみで業務クラスが完成することは無い。

【００４２】

ステップＳ１５又はステップＳ１９の後に、タグ名を業務クラス管理ファイル１５の雛型業務クラス・タグの配下に登録する（ステップＳ２１）。これにより重複する業務クラスを生成することが無くなり、プログラム開発における業務効率化及びプログラムの品質向上に繋がる。そして、全てのタグを処理したか判断する（ステップＳ２３）。もし未処理タグが存在する場合にはステップＳ７に戻る。未処理タグが存在していない場合には処理を終了する。

【００４３】

なお、上の処理では先頭タグに対応して、伝票の前処理、後処理及び伝票処理の雛型ソースプログラムを生成するようにしているが、終了タグに対応して上記雛型ソースプログラムを生成するようにしても良い。

【００４４】

そして、インターフェースマネージャ９は、共通インターフェースファイル群１１の各領域に対応する共通インターフェース雛型ソースファイルを生成し、共通インターフェースファイル群１１の各領域に格納する（ステップＳ２５）。伝票項目域１１１及びＤＢ情報域１１７に格納される共通インターフェース雛型ソースファイルは、伝票項目に対応するため、上で述べた業務クラスの雛型ソースプログラムの生成処理と同じような処理フローにて生成される。すなわち、ＸＭＬデータに含まれるタグのうち伝票項目に対応する未処理タグを１つ選択し、業務クラス管理ファイル１５においてオブジェクトプール・タグの配下に設定されている伝票項目域タグ及びＤＢ情報域タグの配下を参照して、当該タグが未登録であるか確認する。未登録であれば、当該タグに対応する共通インターフェースファイル雛型ソースファイルを伝票項目域１１１及びＤＢ情報域１１７の各々について生成し、さらに業務クラス管理ファイル１５のオブジェクトプール・タグの下にタグを登録する。但し、業務クラス管理ファイル１５に未登録であることを確認の後に伝票項目に対応するタグであるか確認するような処理であってもよい。

【００４５】

また、プロセッサ予約域１１２、クラス間インターフェース領域１１３及びアプリケーション定数域１１５に格納される共通インターフェース雛型ソースファイルは、インターフェースマネージャ９が有する画面入力機能により受け付けた、ユーザからの入力データに基づき生成される。インターフェースマネージャ９は、共通インターフェース雛型ソースファイルを生成すると、対応するタグを、業務クラス管理ファイル１５の各領域に対応するタグの配下に登録する。

【００４６】

なお、上では雛型ソースプログラム（基本的にはそのためのファイル）を生成するような処理フローを示したが、必ずしも雛型ソースプログラムや共通インターフェース雛型ソースファイルを生成する必要は無い。生成すべき業務クラスや共通インターフェースファイルが業務クラス管理ファイル１５に登録されていれば十分な場合もある。このような場合には、業務クラス管理ファイル１５の項目に対応してソースプログラムをコーディングすればよい。そしてコーディングが終了した時点で、再度業務クラス管理ファイル１５の各項目とマッチングをとればよい。

【００４７】

このようにして生成された雛型ソースプログラム及び共通インターフェース雛型ソースファイルについては、業務ロジックに適合するようなコーディングが実施される。すなわち、雛型ソースプログラムは、当該雛型ソースプログラムに共通の事項のみが入力済みとなっており、それぞれにおいて実現すべき機能については、プログラマが別途コーディングする。また、共通インターフェース雛型ソースファイルについても同様に、当該共通インターフェース雛型ソースファイルに共通の事項のみが入力済みとなっており、それぞれにおいて実現すべき機能については、プログラマが別途コーディングする。コーディングされたソース・プログラムはコンパイルされ、実行可能なプログラム・モジュールが生成される。

【００４８】

この際、特定の伝票に対応するＸＭＬデータをステップＳ１５及びステップＳ１９で生成することが決定された業務クラスと当該業務クラスが必要とするオブジェクトプール内のオブジェクトのみで処理できるように、コーディングする。これは以下でも述べるが、特定の伝票に対応するＸＭＬデータを処理する際には、ＸＭＬデータに含まれるタグをベースに、ステップＳ１５及びステップＳ１９で生成することが決定された業務クラスのみがコールされるためである。なお、オブジェクトプールにおけるオブジェクトは、業務クラスにより呼び出されるが、業務クラスによる処理をサポートするものであって、処理の主体は業務クラスにある。従来のようにＸＭＬデータを予め規定された順番でシーケンシャルに処理するためのメイン制御ロジックを用意する必要は無く、業務クラスによりＸＭＬデータのタグで規定される各オブジェクトを実行する。コーディングされたソース・プログラムはコンパイルされ、実行可能なプログラム・モジュールが生成される。

【００４９】

なお、共通インターフェースファイル群１１の各領域に格納された共通インターフェース雛型ソースファイルについては、図４に示すようなオブジェクトを実現すべくコーディングがなされる。なお、図４に示すオブジェクトの構造はオブジェクトプール内では全て同じであり、例えば特定の雛型オブジェクト５００には、データ部５１０とメソッド部５２０とが含まれている。データ部５１０には、データ格納数領域５１１と、１又は複数のデータ領域５１２と、１又は複数のデータ状態値領域５１３と、１又は複数の第１属性データ領域５１４と、・・・、１又は複数の第ｎ（ｎは例えば自然数）属性データ領域５１５と、１又は複数の第１属性状態値領域５１６と、・・・、１又は複数の第ｎ属性状態値領域５１７とが含まれている。また、メソッド部５２０には、項目参照部５２１と項目更新部５２２とデータ状態値参照部５２３とデータ状態値更新部５２４と項目初期化部５２５と属性参照部５２６と属性更新部５２７とデータ格納数取得部５２８とが含まれている。

【００５０】

このようなオブジェクトプールにおける各オブジェクトを使用するために、例えば図５（ａ）乃至図５（ｈ）に示すようなコードを、業務クラスの雛型ソースプログラムに追加する。図５（ａ）には、共通インターフェースファイル群１１の領域別に、図４に示した項目参照部５２１を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトの項目参照部５２１を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．getValue（伝票項目名，配列番号）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分、「伝票項目名」部分及び「配列番号」部分に項目データを特定するための値を設定することにより、適切なオブジェクトによる処理を規定することができる。なお、本実施の形態においては、配列番号が０の場合、先頭レコードを指すようになっており、以下の説明でも同様である。伝票項目域１１１に対応するオブジェクトの項目参照部５２１は、このようなコードに従い、項目データを参照して値を返す。

【００５１】

２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトの項目参照部５２１を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データを参照する処理を規定することができる。

【００５２】

図５（ｂ）には、共通インターフェースファイル群１１の領域別に、図４に示した項目更新部５２２を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトの項目更新部５２２を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．setValue（伝票項目名，データ）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分及

び「伝票項目名」部分に項目データを特定するための値を設定し、且つ「データ」部分に格納すべきデータ（又はその変数）を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトの項目更新部５２２は、このようなコードに従い、項目データを更新する。２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトの項目更新部５２２を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データを更新する処理を規定することができる。

【００５３】

図５（ｃ）には、共通インターフェースファイル群１１の領域別に、図４に示したデータ状態値参照部５２３を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトのデータ状態値参照部５２３を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．getError（伝票項目名，配列番号）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分、「伝票項目名」部分及び「配列番号」部分に項目データを特定するための値を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトのデータ状態値参照部５２３は、このようなコードに従い、項目データの状態値を参照して値を返す。２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトのデータ状態値参照部５２３を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データの状態値を参照する処理を規定することができる。

【００５４】

図５（ｄ）には、共通インターフェースファイル群１１の領域別に、図４に示したデータ状態値更新部５２４を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトのデータ状態値更新部５２４を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．setError（伝票項目名，データ）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分及び「伝票項目名」部分に項目データを特定するための値を設定し、且つ「データ」部分に格納すべき状態値（又はその変数）を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトのデータ状態値更新部５２４は、このようなコードに従い、項目データの状態値を更新する。２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトのデータ状態値更新部５２４を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データの状態値を更新する処理を規定することができる。

【００５５】

図５（ｅ）には、共通インターフェースファイル群１１の領域別に、図４に示したデータ格納数取得部５２８を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトのデータ格納数取得部５２８を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．getSize（伝票項目名）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分及び「伝票項目名」部分に項目を特定するための値を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトのデータ格納数取得部５２８は、このようなコードに従い、データ格納数を取得して値を返す。２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１

１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトのデータ格納数取得部５２８を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データ格納数を参照する処理を規定することができる。

【００５６】

図５（ｆ）には、共通インターフェースファイル群１１の領域別に、図４に示した項目初期化部５２５を呼び出すためのコードの例が示されている。まず１行目には、伝票項目域１１１に対応するオブジェクトの項目初期化部５２５を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．clearValue（伝票項目名）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分及び「伝票項目名」部分に項目を特定するための値を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトの項目初期化部５２５は、このようなコードに従い、項目データの初期化を行う。２行目以降には、順に、ＤＢ情報域１１７、クラス間インターフェース領域１１３、アプリケーション定数域１１５及びプロセッサ予約域１１２の各々について、その領域に対応するオブジェクトの項目初期化部５２５を呼び出すためのコードの例が示されている。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データを、初期化する処理を規定することができる。

【００５７】

図５（ｇ）には、伝票項目域１１１に対応するオブジェクトの属性参照部５２６（図４）を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．getAttributeVal（伝票項目名，配列番号，属性名）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分、「伝票項目名」部分、「配列番号」部分及び「属性名」部分に属性データを特定するための値を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトの属性参照部５２６は、このようなコードに従い、属性データを参照して値を返す。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データの属性値を、参照する処理をコーディングすることができる。

【００５８】

図５（ｈ）には、伝票項目域１１１に対応するオブジェクトの属性更新部５２７（図４）を呼び出すためのコードの例として、「オブジェクトプールインスタンス変数．setAttributeVal（伝票項目名，配列番号，属性名，属性値）」というコードが示されている。例えばプログラマは「オブジェクトプールインスタンス変数」部分、「伝票項目名」部分、「配列番号」部分及び「属性名」部分に属性データを特定するための値を設定し、且つ「属性値」部分に格納すべき属性値（又はその変数）を設定することにより、適切なオブジェクトによる処理を規定することができる。伝票項目域１１１に対応するオブジェクトの属性更新部５２７は、このようなコードに従い、属性データの更新を行う。例えばプログラマは、このようなコードを業務クラスの雛型ソースプログラムに追加することにより、オブジェクトプールに格納されている項目データの属性値を更新する処理を規定することができる。

【００５９】

２．業務クラス実行時の処理

業務クラスを実行する際における、本発明の一実施の形態に係るコンピュータ・システムの機能ブロック図を図６に示す。コンピュータ・ネットワーク２０１には、１又は複数の端末装置２０３と、本実施の形態における主たる処理を実施するサーバ２０５とが接続されている。端末装置２０３は、ＸＭＬ対応のブラウザを有している。また、サーバ２０５は、端末装置２０３のブラウザとのインターフェースとなるサーブレット２０５１と、

入力画面用ＸＭＬデータ２０５４や出力画面用ＸＭＬデータ２０５５を格納するメモリ領域２０６０と、入力画面のための雛型ＸＭＬデータ２０６２とＸＳＬデータ２０６３とスキーマ２０６４とを格納する第１ＸＭＬデータ格納部２０６１と、ＸＭＬデータを解析して対応する業務クラスを呼び出す処理などを実施するプロセッサ２０７１と、業務クラス管理ファイル１５と、業務クラス群がロードされるメモリ領域２０８０と、出力画面のための雛型ＸＭＬデータ２０９２とＸＳＬデータ２０９３とスキーマ２０９４とを格納する第２ＸＭＬデータ格納部２０９１と、共通インターフェースファイルがロードされ且つインデックス情報２１０２を保持するメモリ領域２１０１（オブジェクトプール）と、実行可能な業務クラスのファイルが格納された業務クラス格納部２１１１と、オブジェクトプールにおけるオブジェクトを構成するためのクラスファイルであって且つ実行可能な共通インターフェースファイルが格納された共通インターフェースファイル格納部２１１２と、初期設定ファイル１１９とが含まれる。

【００６０】

サーブレット２０５１には、端末装置２０３のブラウザからのリクエストに応答するための処理を実施するリクエスト応答部２０５２と、端末装置２０３のブラウザから受信した選択又は入力データに対応してＸＭＬデータ（入力画面用ＸＭＬデータ２０５４）を生成及び出力するＸＭＬ生成部２０５６と、プロセッサ２０７１から処理結果を含むＸＭＬデータ（出力画面用ＸＭＬデータ２０５５）を受信し、端末装置２０３のブラウザに出力する結果出力部２０５３とが含まれる。

【００６１】

プロセッサ２０７１には、メモリ領域２１０１を初期化（すなわち、共通インターフェースファイルを共通インターフェースファイル格納部２１１２からロードし且つインデックス情報２１０２を生成）したり、業務クラスを業務クラス格納部２１１１からメモリ領域２０８０にロードするなどの初期処理を実施する初期処理部２０７２と、入力画面用ＸＭＬデータ２０５４を処理するＸＭＬデータ処理部２０７３と、メモリ領域２１０１の伝票項目域２１０３に格納されたデータ等と第２ＸＭＬデータ格納部２０９１に格納された雛型ＸＭＬデータ２０９２とに基づき出力画面用ＸＭＬデータ２０５５を生成する出力ＸＭＬ生成部２０７４とが含まれる。なお、メモリ領域２１０１のＤＢ情報域２１０７は例えばハッシュ・テーブル形式でデータを格納するようになっており、インデックス情報２１０２にデータの格納場所に関するデータが格納される。

【００６２】

次に図７乃至図１３を用いて図６に示したコンピュータ・システム２００の処理フローを説明する。まず、サーバ２０５におけるサーブレット２０５１のリクエスト応答部２０５２は、端末装置２０３のブラウザから特定の入力画面データの要求を受信する（ステップＳ３１）。そうすると、リクエスト応答部２０５２は、特定の入力画面用の雛型ＸＭＬデータ２０６２及び当該雛型ＸＭＬデータ２０６２に対応するＸＳＬデータ２０６３を第１ＸＭＬデータ格納部２０６１から読み出し、端末装置２０３のブラウザに送信する（ステップＳ３３）。なお、実際の処理では、リクエスト応答部２０５２は、まず雛型ＸＭＬデータ２０６２を端末装置２０３のブラウザに送信しておき、端末装置２０３のブラウザからの雛型ＸＭＬデータ２０６２に対応するＸＳＬデータの送信要求を受信することにより、ＸＳＬデータ２０６３を端末装置２０３のブラウザに送信するが、本実施の形態の主旨ではないため、上の説明及び以下の同様の処理についての説明では、ＸＭＬデータとＸＳＬデータとが併せて送信されるように記載を簡略化している。

【００６３】

端末装置２０３のブラウザは、サーバ２０５から特定の入力画面用の雛型ＸＭＬデータ２０６２及び雛型ＸＭＬデータ２０６２に対応するＸＳＬデータ２０６３を受信すると、受信したデータに基づきＨＴＭＬデータを生成し、表示装置に表示する。例えば、図８に示すような画面が端末装置２０３の表示装置に表示される。図８の例は、入金取引の入力画面であって、銀行名の入力又は選択欄６０１と、支店名の入力又は選択欄６０２と、口座種別の選択欄（ラジオボタン）６０３と、口座番号の入力欄６０４と、取引金額の入力

欄６０５と、確認ボタン６０６と、戻るボタン６０７とが含まれる。すなわち、伝票名及び伝票項目に対応するタグが、雛型ＸＭＬデータ２０６２には含まれている。

【００６４】

端末装置２０３のユーザは、端末装置２０３を操作してデータを選択又は入力して確認ボタン６０６をクリックする。そうすると、端末装置２０３のブラウザは、例えばＨＴＴＰ（Hyper Text Transfer Protocol）のＰＯＳＴメソッドにより入力又は選択データをサーバ２０５に送信する。サーバ２０５におけるサーブレット２０５１のＸＭＬ生成部２０５６は、端末装置２０３のブラウザから入力又は選択データを受信し、一旦メモリに格納する（ステップＳ３５）。そして、ＸＭＬ生成部２０５６は、入力又は選択データを解析し、入力画面用ＸＭＬデータ２０５４を生成し、メモリ領域２０６０に格納する（ステップＳ３７）。なお、ＸＭＬ生成部２０５６は、例えばＸＳＬデータ２０６３を参照して入力又は選択データを分節し、雛型ＸＭＬデータ２０６２に埋め込むことにより、入力画面用ＸＭＬデータ２０５４を生成する。また、スキーマ２０６４を用いて入力画面用ＸＭＬデータ２０５４の妥当性を確認する。そして、ＸＭＬ生成部２０５６は、プロセッサ２０７１を呼び出し、入力画面用ＸＭＬデータ２０５４をプロセッサ２０７１に出力する（ステップＳ３９）。

【００６５】

入力画面用ＸＭＬデータ２０５４の一例を図９に示す。図９の例では、伝票名に対応する入金取引入力タグと、２００という実データを挟持する銀行名タグ、１２１という実データを挟持する支店名タグ、０１という実データを挟持する口座種別タグ、１２３４５６７８という実データを挟持する口座番号タグと、１００００という実データを挟持する取引金額タグとが含まれる。

【００６６】

一方、サーバ２０５におけるプロセッサ２０７１の初期処理部２０７２は、例えば端末装置２０３とのセッションが張られた時に、オブジェクトプールの初期化を実施する（ステップＳ４１）。例えば業務クラス管理ファイル１５を参照して、オブジェクトプールとして列挙されたクラスに係る共通インターフェースファイルを共通インターフェースファイル格納部２１１２からメモリ領域２１０１にロードする。なお、共通インターフェースファイル格納部２１１２には、共通インターフェースファイル群１１の各領域に格納された共通インターフェース雛型ソースファイルに基づき作成された、実行可能な共通インターフェースファイルが格納されている。

【００６７】

また、初期処理部２０７２は、業務クラス管理ファイル１５を参照して、業務クラス格納部２１１１に格納された全ての業務クラスのファイルをメモリ領域２０８０にロードする（ステップＳ４３）。そして、例えばプロセス初期処理クラスを呼び出し、実行させる（ステップＳ４５）。プロセス初期処理クラスは、図３のステップＳ５において生成された雛型ソースプログラムを基に適切にコーディングされたクラスである。また、伝票項目域２１０３等を初期化する（ステップＳ４７）。この際、アプリケーション定数域２１０６には、初期設定ファイル１１９に基づく項目データがプロセス初期処理クラスにより格納される。

【００６８】

そして、ステップＳ３９でサーブレット２０５１が入力画面用ＸＭＬデータ２０５４を出力したことに応じて、プロセッサ２０７１のＸＭＬデータ処理部２０７３は、入力画面用ＸＭＬデータ２０５４を受信する（ステップＳ４９）。プロセッサ２０７１のＸＭＬデータ処理部２０７３は、入力画面用ＸＭＬデータ２０５４を解析及び文節して、実データ（入力又は選択データ）を伝票項目域２１０３の対応する領域（オブジェクト）に格納する（ステップＳ５１）。図９のような入力画面用ＸＭＬデータ２０５４を処理する場合には、銀行名のデータ格納領域、支店名のデータ格納領域、口座種別のデータ格納領域、口座番号のデータ格納領域、及び取引金額のデータ格納領域に、実データを格納する。

【００６９】

プロセッサ２０７１のＸＭＬデータ処理部２０７３は、入力画面用ＸＭＬデータ２０５４の伝票名に対応する前処理クラス（ここでは入金取引入力前処理クラス２０８１）を呼び出し、業務処理を実行させる（ステップＳ５３）。また、終了タグ以外のタグ毎に（ステップＳ５５：Ｎｏルート）、業務クラス（銀行名クラス２０８２、支店名クラス２０８３、口座種別クラス２０８４、口座番号クラス２０８５、取引金額クラス２０８６）をそれぞれ呼び出し、業務処理を実行させる（ステップＳ５７）。そして、入力画面ＸＭＬデータ２０５４の終了タグが検出されると（ステップＳ５５：Ｙｅｓルート）、伝票名に対応する伝票処理クラス（入金取引入力伝票処理クラス２０８７）を呼び出し、業務処理を実施させる（ステップＳ５９）。そして、伝票名に対応する後処理クラス（入力取引入力後処理クラス２０８８）を呼び出し、実行させる（ステップＳ６１）。

【００７０】

これら呼び出された業務クラスは、メモリ領域２１０１（オブジェクトプール）の各領域に属するオブジェクトに格納されているデータを参照又は更新する等、適宜利用して処理を実行する。なお、業務クラスにおける処理において、手数料、合計金額、残高、取扱店番、取扱支店名、取引日、時刻といったデータも別途生成されて、例えば伝票項目域２１０３に属するオブジェクト及びＤＢ情報域２１０７に属するオブジェクトに格納される。

【００７１】

図１０Ａ乃至図１０Ｄに、オブジェクトプールを用いた処理の概念図を示す。なお、図１０Ａ乃至図１０Ｄには、処理の概要を説明するために必要な構成要素のみを示しており、その他、実際の処理上必要となる構成要素を省略している場合がある。図１０Ａには、伝票項目域２１０３に属するオブジェクトを用いた処理の例が示されている。図１０Ａの例では、例えば端末装置２０３の表示装置に表示されているブラウザ画面１００１には、入力項目１００２及び入力項目１００３が含まれている。そこでユーザが端末装置２０３を操作して入力項目１００２及び入力項目１００３の少なくともいずれかにデータを入力し、図示しない確認ボタン等をクリックする。そうすると入力項目１００２及び１００３に対応する入力データはサーバ２０５に送信され、サーブレット２０５１によりＸＭＬデータ１００４（図６における入力画面用ＸＭＬデータ２０５４に相当）が生成される。ＸＭＬデータ１００４は、例えばＸＭＬ文１００５に示したような内容を含んでいる。すなわち、入力項目の属性１の値及び属性２の値と、ユーザから入力された実データとが含まれている。プロセッサ２０７１は、ＸＭＬデータ１００４のタグ名に基づき、オブジェクトプール（ここでは伝票項目域２１０３）の中から入力項目に対応するオブジェクトを特定し、属性１の値及び属性２の値とユーザから入力された実データとを当該オブジェクトに出力する。そして、例えばオブジェクト１００８はデータ領域１０１０に実データを格納し、属性領域１０１１に属性１の値及び属性２の値を格納する。図１０Ａに示した例のように属性が複数定義されている場合には、複数の属性の値が属性領域１０１１に格納される。なお、プロセッサ２０７１はＸＭＬデータ１００４のタグに基づき業務クラス１０１２をコールする。業務クラス１０１２は、必要に応じてオブジェクトプール２１０３内のオブジェクト（例えばオブジェクト１００８）を呼び出し、データ領域１０１０と属性領域１０１１とに格納されたデータの参照処理や更新処理を行うことにより、伝票処理を実施する。

【００７２】

図１０Ｂには、ＤＢ情報域２１０７に属するオブジェクトを用いた処理の例が示されている。例えばプロセッサ２０７１からコールされた業務クラス１０１３は、オブジェクトプール（ここではＤＢ情報域２１０７）のオブジェクト１０２１に対して、ＤＢ１０２４から抽出したレコードのデータを出力する。そして、オブジェクト１０２１は、データ領域１０２３にデータを格納する。業務クラス１０１３がＤＢ１０２４から複数のレコードを抽出していた場合には、当該複数のレコードのデータを各々データ領域１０２３に格納する。なお、データ領域１０２３は、例えばハッシュ・テーブルである。

【００７３】

図１０Ｃには、アプリケーション定数域２１０６に属するオブジェクトを用いた処理の例が示されている。上でも述べたが、プロセッサ２０７１（初期処理部２０７２）は、オブジェクトプールの初期化を実施する際、プロセス初期処理クラス１０１４を呼び出す。プロセス初期処理クラス１０１４は、初期設定ファイル１１９を読み込み、アプリケーション定数項目データをアプリケーション定数域２１０６のオブジェクトに格納する。例えばオブジェクト１０３１及びオブジェクト１０３２にデータが格納される。そして例えば業務クラス１０１５は、必要に応じてオブジェクト１０３１やオブジェクト１０３２を呼び出し、格納されているデータを参照することにより、伝票処理を実施する。

【００７４】

図１０Ｄには、クラス間インターフェース領域２１０５に属するオブジェクトを用いた処理の例が示されている。図１０Ｄでは、業務クラスであるクラスＡ（１０４５）からクラスＢ（１０４６）にデータを渡し、クラスＢ（１０４６）からクラスＡ（１０４５）にデータを返すという処理を行う場合の例が示されている。まず、クラスＡ（１０４５）は、矢印１０４７に示すように、オブジェクトプール（ここではクラス間インターフェース領域２１０５）内のクラスＢのインターフェース項目域に属するオブジェクト１０４１にデータを出力する。そしてオブジェクト１０４１はデータを入力領域１０４２に格納する。次にクラスＢ（１０４６）は、矢印１０４９に示すように、入力領域１０４２に格納されたデータを参照する。そしてクラスＢ（１０４６）は、矢印１０５０に示すように、オブジェクト１０４１に対し、参照データに基づき所定の処理を行うことにより生成されたデータを出力する。オブジェクト１０４１は、クラスＢ（１０４６）から受信したデータを出力領域１０４３に格納する。そして、クラスＡ（１０４５）は、矢印１０４８に示すように、出力領域１０４３に格納されたデータを参照する。このように、業務クラス間では直接データのやり取りを行うことなく、クラス間インターフェース領域２１０５に属するオブジェクトを介してデータのやりとりを行う。これにより、各業務クラスを他の業務クラスを考慮することなく作成することができるため、プログラムの作成効率が向上する。

【００７５】

なお、各業務クラスの具体的な処理内容については、業務ロジックに即した形でコーディングされ、上でも述べたが、特定の伝票の処理は、入力画面用ＸＭＬデータ２０５４に基づきＸＭＬデータ処理部２０７３により呼び出される業務クラスによる処理で過不足無く完結するように構成されている。また、ＸＭＬデータ処理部２０７３は、入力画面用ＸＭＬデータ２０５４に含まれるタグに、実際に存在しないような業務クラスに対応するタグが含まれる場合には、業務クラスの呼び出しに失敗する。そのような場合には、適切な処理が実施できない又は不適切な入力が発生しているので、エラーをサーブレット２０５１に出力して、処理を終了させる。

【００７６】

全ての処理が完了すると、後処理クラス（入金取引入力後処理クラス２０８８）は、第２ＸＭＬデータの生成要求（すなわち、出力画面用ＸＭＬの文書名を含む出力伝票生成要求）を、プロセッサ２０７１の出力ＸＭＬ生成部２０７４に出力する。処理は、図７の端子Ａを介して図１１の処理に遷移する。

【００７７】

出力ＸＭＬ生成部２０７４は、出力画面用ＸＭＬデータの生成要求に応じて、第２ＸＭＬデータ格納部２０９１から該当する雛型ＸＭＬデータ２０９２を取得する（ステップＳ６３）。この際、出力画面用ＸＭＬデータに対応するスキーマ２０９４も併せて取得する。また、メモリ領域２１０１の伝票項目域２１０３に属するオブジェクトから、読み出した雛型ＸＭＬデータ２０９２に含まれるタグに対応するデータ項目の実データを受け取り、雛型ＸＭＬデータ２０９２に埋め込むことにより、出力画面用ＸＭＬデータ２０５５を生成し、メモリ領域２０６０に格納する。この際、スキーマ２０９４を用いて出力画面用ＸＭＬデータ２０５５の妥当性を確認する。そして、当該出力画面用ＸＭＬデータ２０５５をサーブレット２０５１に出力する（ステップＳ６５）。

【００７８】
出力画面用ＸＭＬデータ２０５５の一例を図１２に示す。図１２の例では、スタイルシート（出力画面．ｘｓｌ）を指定するタグと、伝票名を表すタグ（入金取引結果タグ）と、伝票項目を表すタグ（取引日タグ、時刻タグ、銀行名タグ、支店名タグ、口座種別タグ、口座番号タグ、取引金額タグ、手数料タグ、合計金額タグ、残高タグ、取扱店番タグ、取扱支店名タグ）と、それぞれのタグに対応する実データとが含まれる。
【００７９】
なお、プロセッサ２０７１側でプロセスが終了とは判断されなければ端子Ｂを介して図７のステップＳ４７に戻る。一方、プロセスが終了と判断されれば、プロセス終了クラスを呼び出し、実行させる（ステップＳ７３）。プロセス終了クラスは、図３のステップＳ５において生成された雛型ソースプログラムを基に、適切にコーディングされたクラスである。
【００８０】
サーブレット２０５１の結果出力部２０５３は、プロセッサ２０７１から出力画面用ＸＭＬデータ２０５５を受け取り（ステップＳ６７）、当該出力画面用ＸＭＬデータ２０５５及び第２ＸＭＬデータ格納部２０９１から読み出されたＸＳＬデータ２０９３を端末装置２０３に出力する（ステップＳ６９）。端末装置２０３のブラウザは、サーバ２０５から出力画面用ＸＭＬデータ２０５５及びＸＳＬデータ２０９３を受信し、ＨＴＭＬデータを生成して表示装置に表示する。例えば図１３のような画面を表示する。図１３の例では、出力伝票名（入金取引（結果））、取引日、時刻、銀行名、支店名、口座種別、口座番号、取引金額、手数料、合計金額、残高、取扱店番、取扱支店名の各表示欄が設けられている。さらに、確認ボタン１３０１とメニューに戻る初画面ボタン１３０２も設けられる。
【００８１】
このように、業務クラスを適切に呼び出し、処理を実施させることにより、伝票を含む帳票の処理が適切に行われる。なお、サーブレット２０５１及びプロセッサ２０７１とについては全システム共通で用いることができ、業務（伝票）ごとに異なる部分については業務クラスを作成する。このようなプログラム開発手法を採用することにより、プログラムの再利用、開発の分業及び開発期間の短縮を図ることができる。さらに、業務クラスを機能を限定して他の業務クラスから分離して作成するので、プログラムの品質向上に繋がる。
【００８２】
以上本発明の一実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、図１及び図６に示した機能ブロック図は、本実施の形態を説明する上で分離した機能であり、必ずしもプログラムのモジュールに対応するものではない。
【００８３】
（付記１）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断するステップと、
特定された前記業務クラスの項目が未登録である場合には、前記業務クラス管理部に当該特定された前記業務クラスの項目を登録するステップと、
を含み、コンピュータにより実行される情報処理方法。
【００８４】
（付記２）
特定された前記業務クラスの項目が未登録である場合には、当該特定された前記業務クラスのための雛型ソースプログラム・データを生成し、ソースプログラム格納部に格納するステップ
をさらに含む付記１記載の情報処理方法。

【００８５】
（付記３）
ＨＴＭＬファイル格納部から帳票画面のためのＨＴＭＬファイルを読み出し、所定のルールに従って前記帳票画面に対応するＸＭＬデータを生成し、前記ＸＭＬデータ格納部に格納するステップ
をさらに含む付記１記載の情報処理方法。
【００８６】
（付記４）
前記業務クラス特定ステップが、
前記帳票画面に対応するＸＭＬデータの開始又は終了タグに対応して、前処理クラス、後処理クラス及び帳票処理クラスを特定するステップ
を含む付記１記載の情報処理方法。
【００８７】
（付記５）
ユーザの指示に応じて帳票画面に対応するＨＴＭＬファイルを生成し、前記ＨＴＭＬファイル格納部に格納するステップ
をさらに含む付記３記載の情報処理方法。
【００８８】
（付記６）
前記ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータに含まれるタグにより帳票項目を特定するステップと、
生成すべき帳票項目格納オブジェクトが登録された帳票項目格納オブジェクト管理部を参照して、特定された前記帳票項目が未登録であるか判断するステップと、
特定された前記帳票項目が未登録である場合には、前記帳票項目格納オブジェクト管理部に当該特定された前記帳票項目を登録するステップと、
をさらに含む付記１記載の情報処理方法。
【００８９】
（付記７）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断するステップと、
特定された前記業務クラスの項目が未登録である場合には、当該特定された前記業務クラスのための雛型ソースプログラム・データを生成し、ソースプログラム格納部に格納するステップと、
を含み、コンピュータにより実行される情報処理方法。
【００９０】
（付記８）
帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する業務クラス特定ステップと、
予め定義され且つメモリにロードされた業務クラスの中で、特定された前記業務クラスを呼び出す呼出ステップと、
を含み、コンピュータにより実行される情報処理方法。
【００９１】
（付記９）
前記業務クラス特定ステップが、
帳票画面に係る帳票と前記ＸＭＬデータに含まれるタグとに対応する前処理クラス、後処理クラス及び帳票処理クラスを特定するステップ
を含む付記８記載の情報処理方法。

【００９２】
（付記１０）
前記帳票画面に対する入力又は選択データを、予め定義され且つメモリにロードされた帳票項目格納オブジェクトに出力し、当該帳票項目格納オブジェクトにより前記帳票画面に対する入力又は選択データをメモリに格納するステップと、
呼び出された前記業務クラスと前記帳票項目格納オブジェクトとがデータをやり取りすることにより処理を実施する処理実施ステップと、
をさらに含む付記８記載の情報処理方法。
【００９３】
（付記１１）
前記処理実施ステップが、
第１の業務クラスから第２の業務クラスへデータを渡す場合に、前記第１の業務クラスが、予め定義され且つメモリにロードされたクラス間インターフェース・オブジェクトにデータを出力するステップと、
前記第２の業務クラスが前記クラス間インターフェース・オブジェクトを参照して、前記データを受け取るステップと、
を含む付記１０記載の情報処理方法。
【００９４】
（付記１２）
帳票画面に対する入力又は選択データを帳票画面を表示した装置から受信し、当該帳票画面に対する入力又は選択データ及び対応するタグを含むＸＭＬデータを生成するステップ
をさらに含む付記８記載の情報処理方法。
【００９５】
（付記１３）
呼び出された前記業務クラスにより出力要求を受けた場合、前記帳票項目格納オブジェクトに保持されたデータを用いて出力用ＸＭＬデータを生成し、メモリに格納するステップと、
前記メモリに格納された前記出力用ＸＭＬデータを前記帳票画面を表示した装置に出力するステップと、
をさらに含む付記１０記載の情報処理方法。
【００９６】
（付記１４）
特定された前記業務クラスが存在しない場合、エラー情報を生成し、出力するステップ
をさらに含む付記８記載の情報処理方法。
【００９７】
（付記１５）
前記帳票画面に係る帳票に対応して当該帳票全体の処理シーケンスが規定されておらず、呼び出された前記業務クラスの処理にて当該帳票に対する処理が完了するように、前記業務クラスの各々が構成されていることを特徴とする付記８記載の情報処理方法。
【００９８】
（付記１６）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断するステップと、
特定された前記業務クラスの項目が未登録である場合には、前記業務クラス管理部に当該特定された前記業務クラスの項目を登録するステップと、
をコンピュータに実行させるプログラム。
【００９９】

（付記１７）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する業務クラス特定ステップと、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断するステップと、
特定された前記業務クラスの項目が未登録である場合には、当該特定された前記業務クラスのための雛型ソースプログラム・データを生成し、ソースプログラム格納部に格納するステップと、
をコンピュータに実行させるプログラム。
【０１００】
（付記１８）
帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する業務クラス特定ステップと、
予め定義され且つメモリにロードされた業務クラスの中で、特定された前記業務クラスを呼び出す呼出ステップと、
をコンピュータに実行させるプログラム。
【０１０１】
（付記１９）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する手段と、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断する手段と、
特定された前記業務クラスの項目が未登録である場合には、前記業務クラス管理部に当該特定された前記業務クラスの項目を登録する手段と、
を有するコンピュータ・システム。
【０１０２】
（付記２０）
ＸＭＬデータ格納部に格納され且つ帳票画面に対応するＸＭＬデータを解析し、前記ＸＭＬデータに含まれるタグに対応する業務クラスを特定する手段と、
生成すべき業務クラスの項目が登録された業務クラス管理部を参照して、特定された前記業務クラスの項目が未登録であるか判断する手段と、
特定された前記業務クラスの項目が未登録である場合には、当該特定された前記業務クラスのための雛型ソースプログラム・データを生成し、ソースプログラム格納部に格納する手段と、
を有するコンピュータ・システム。
【０１０３】
（付記２１）
帳票画面に対する入力又は選択データに対応するタグを含むＸＭＬデータをメモリに格納し、当該ＸＭＬデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定する手段と、
予め定義され且つメモリにロードされた業務クラスの中で、特定された前記業務クラスを呼び出す手段と、
を有するコンピュータ・システム。
【図面の簡単な説明】
【０１０４】
【図１】業務クラス作成時における機能ブロック図である。
【図２】業務クラス管理ファイルに格納されるデータの一例を示す図である。
【図３】業務クラス作成時における処理フローを示す図である。
【図４】オブジェクトの模式図である。

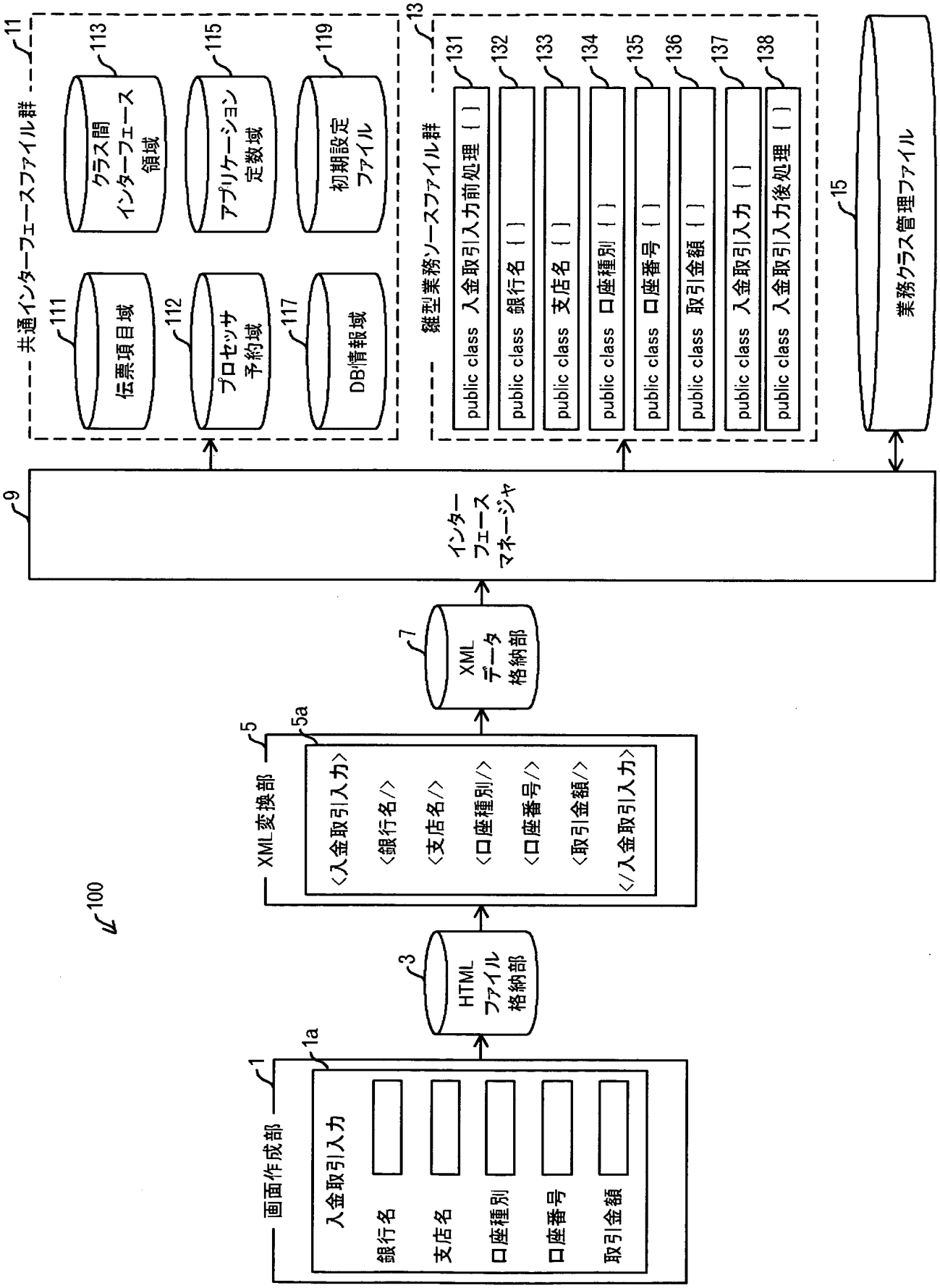
【図５】オブジェクトを使用するためのコードの例を示す図である。
【図６】業務クラス実行時における機能ブロック図である。
【図７】業務クラス実行時における第１の処理フローを示す図である。
【図８】入力画面例を示す図である。
【図９】入力画面用ＸＭＬデータの一例を示す図である。
【図１０Ａ】オブジェクトプールを用いた処理の第１の例を示す図である。
【図１０Ｂ】オブジェクトプールを用いた処理の第２の例を示す図である。
【図１０Ｃ】オブジェクトプールを用いた処理の第３の例を示す図である。
【図１０Ｄ】オブジェクトプールを用いた処理の第４の例を示す図である。
【図１１】業務クラス実行時における第２の処理フローを示す図である。
【図１２】出力画面用ＸＭＬデータの一例を示す図である。
【図１３】出力画面例を示す図である。

【符号の説明】

【０１０５】

１　画面作成部　　３　ＨＴＭＬファイル格納部　　５　ＸＭＬ変換部
７　ＸＭＬデータ格納部　　９　インターフェースマネージャ
１１　共通インターフェースファイル群　　１３　雛型業務ソースファイル群
１５　業務クラス管理ファイル
２０１　コンピュータ・ネットワーク　　２０３　端末装置　　２０５　サーバ
２０５１　サーブレット　　２０６１　第１ＸＭＬデータ格納部
２０７１　プロセッサ　　２１１１　業務クラス格納部
２０９１　第２ＸＭＬデータ格納部
２０６０，２０８０，２１０１　メモリ領域

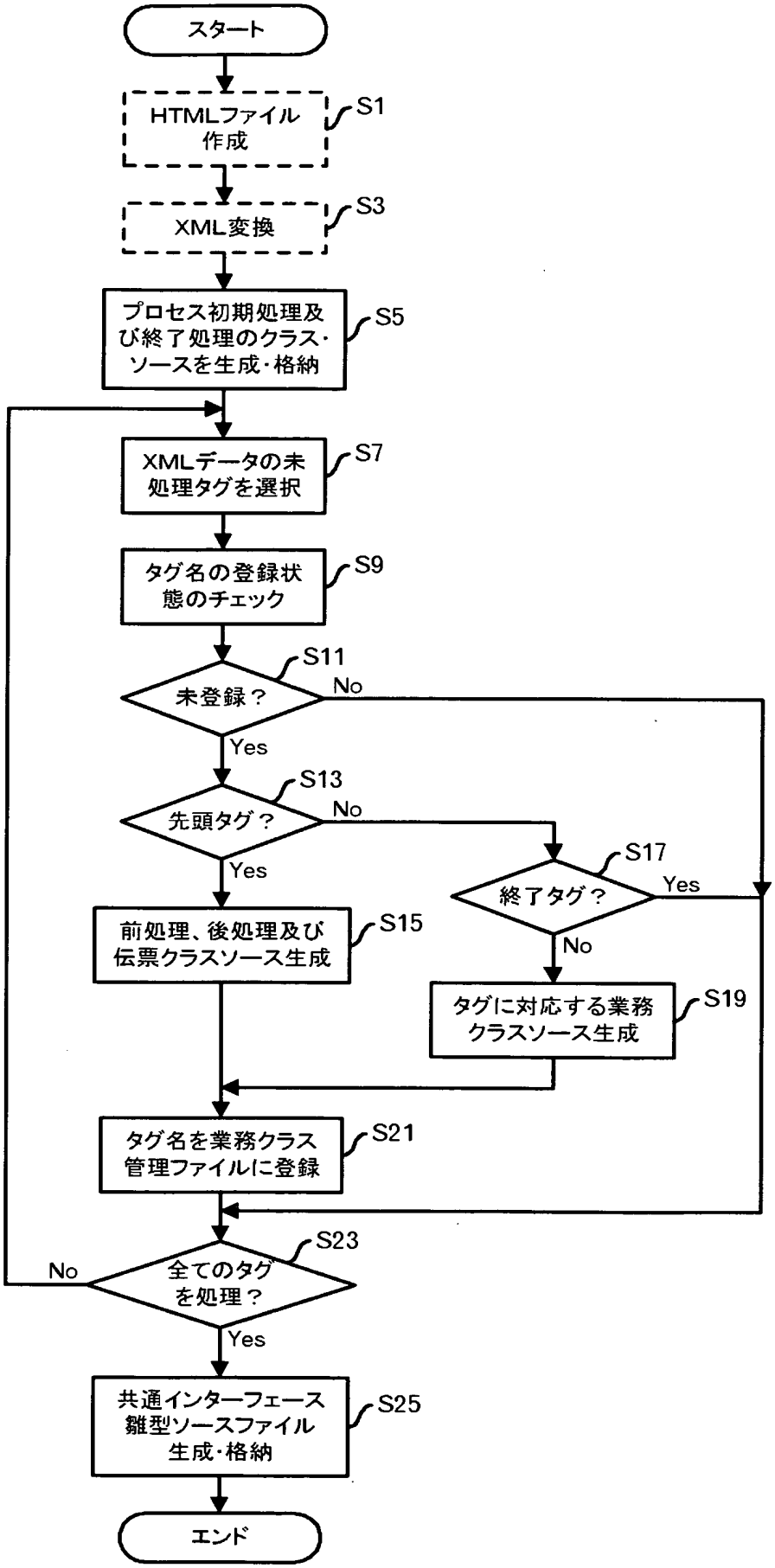
【書類名】図面
【図１】


100
画面作成部
1
1a
入金取引入力
銀行名
支店名
口座種別
口座番号
取引金額
HTML
ファイル
格納部
3
XML変換部
5
5a
〈入金取引入力〉
〈銀行名/〉
〈支店名/〉
〈口座種別/〉
〈口座番号/〉
〈取引金額/〉
〈/入金取引入力〉
XML
データ
格納部
7
インター
フェース
マネージャ
9
共通インターフェースファイル群
11
伝票項目域
111
プロセッサ
予約域
112
クラス間
インターフェース
領域
113
DB情報域
117
アプリケーション
定数域
115
初期設定
ファイル
119
雛型業務ソースファイル群
13
public class 入金取引入力前処理 { }
131
public class 銀行名 { }
132
public class 支店名 { }
133
public class 口座種別 { }
134
public class 口座番号 { }
135
public class 取引金額 { }
136
public class 入金取引入力 { }
137
public class 入金取引入力後処理 { }
138
業務クラス管理ファイル
15

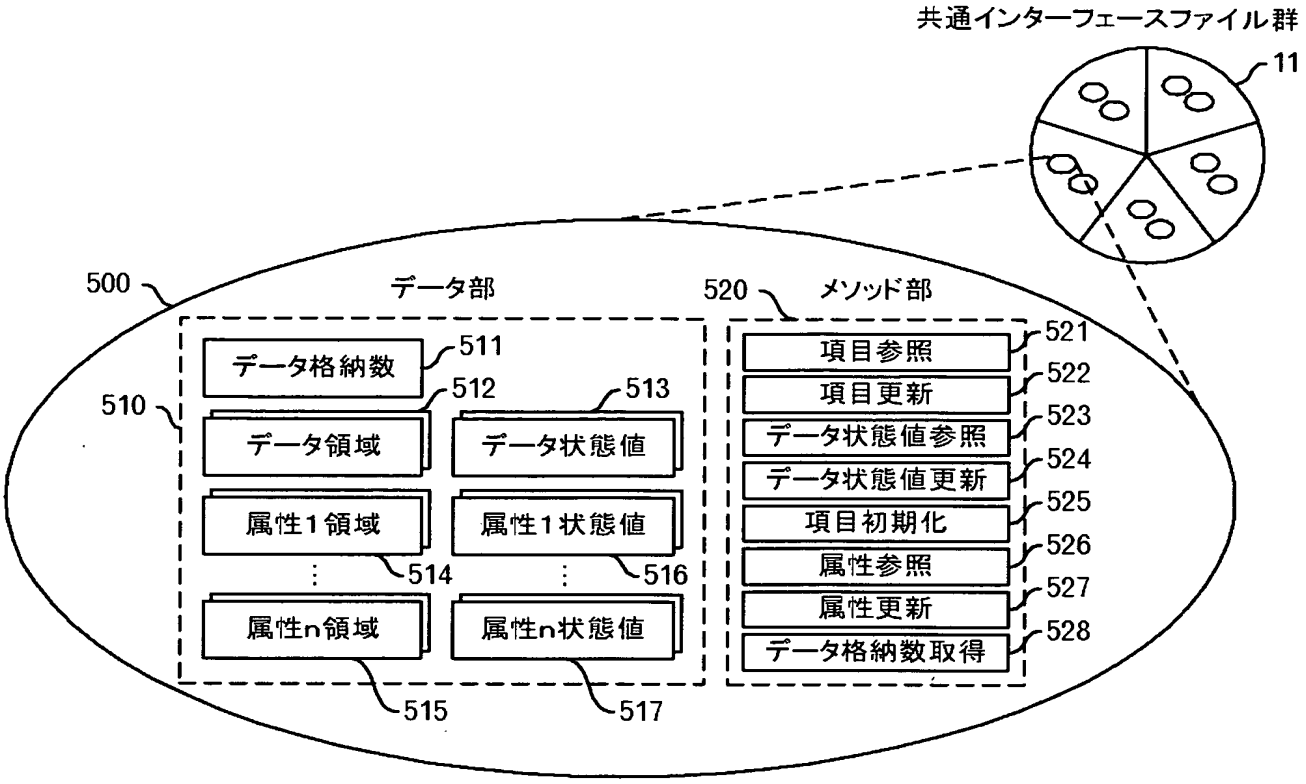
【図２】

```
<?xml version="1.0" encoding="Shift_JIS"?>
- <資産管理>
 - <オブジェクトプール>
  + <プロセッサ予約域>
  - <伝票項目域>
      <入金取引入力　ルート判定="ルート" 登録日時="2003/7/23 10:54:40"/>
      <取引金額　登録日時="2003/7/23 10:54:40" />
      <口座番号　登録日時="2003/7/23 10:54:40" />
      <口座種別　登録日時="2003/7/23 10:54:40" />
      <支店名　登録日時="2003/7/23 10:54:40" / >
      <銀行名　登録日時="2003/7/23 10:54:40" />
    </伝票項目域>
    <DB情報域 />
    <アプリ定数項目域 />
  + <クラス間インタフェース項目域>
  </オブジェクトプール >
- <雛形業務クラス>
    <USR_取引金額　登録日時="2003/7/23 10:54:40" />
    <USR_口座番号　登録日時="2003/7/23 10:54:40" />
    <USR_口座種別　登録日時="2003/7/23 10:54:40" />
    <USR_支店名　登録日時="2003/7/23 10:54:40" />
    <USR_銀行名　登録日時="2003/7/23 10:54:40" />
    <USR_入金取引入力前処理　登録日時="2003/7/23 10:54:40" />
    <USR_入金取引入力後処理　登録日時="2003/7/23 10:54:40" />
    <USR_入金取引入力　登録日時="2003/7/23 10:54:40" />
  </雛形業務クラス>
</資産管理>
```

【図３】


スタート
HTMLファイル作成
S1
XML変換
S3
プロセス初期処理及び終了処理のクラス・ソースを生成・格納
S5
XMLデータの未処理タグを選択
S7
タグ名の登録状態のチェック
S9
未登録？
S11
No
Yes
先頭タグ？
S13
No
Yes
前処理、後処理及び伝票クラスソース生成
S15
終了タグ？
S17
Yes
No
タグに対応する業務クラスソース生成
S19
タグ名を業務クラス管理ファイルに登録
S21
全てのタグを処理？
S23
No
Yes
共通インターフェース雛型ソースファイル生成・格納
S25
エンド

【図４】


共通インターフェースファイル群
11
500
データ部
520
メソッド部
510
データ格納数
511
データ領域
512
データ状態値
513
属性1領域
514
属性1状態値
516
属性n領域
515
属性n状態値
517
項目参照
521
項目更新
522
データ状態値参照
523
データ状態値更新
524
項目初期化
525
属性参照
526
属性更新
527
データ格納数取得
528

【図５】

(a)

オブジェクトプールインスタンス変数. getValue(伝票項目名, 配列番号)
オブジェクトプールインスタンス変数. getDBValue(DB項目名, 配列番号)
オブジェクトプールインスタンス変数. getAphandleValue(クラス間インターフェース項目名, 配列番号)
オブジェクトプールインスタンス変数. getApconstValue(アプリケーション定数項目名, 配列番号)
オブジェクトプールインスタンス変数. getProcessorValue(プロセッサ予約項目名, 配列番号)

(b)

オブジェクトプールインスタンス変数. setValue(伝票項目名, データ)
オブジェクトプールインスタンス変数. setDBValue(DB項目名, データ)
オブジェクトプールインスタンス変数. setAphandleValue(クラス間インターフェース項目名, データ)
オブジェクトプールインスタンス変数. setApconstValue(アプリケーション定数項目名, データ)
オブジェクトプールインスタンス変数. setProcessorValue(プロセッサ予約項目名, データ)

(c)

オブジェクトプールインスタンス変数. getError(伝票項目名, 配列番号)
オブジェクトプールインスタンス変数. getDBError(DB項目名, 配列番号)
オブジェクトプールインスタンス変数. getAphandleError(クラス間インターフェース項目名, 配列番号)
オブジェクトプールインスタンス変数. getApconstError(アプリケーション定数項目名, 配列番号)
オブジェクトプールインスタンス変数. getProcessorError(プロセッサ予約項目名, 配列番号)

(d)

オブジェクトプールインスタンス変数. setError(伝票項目名, データ)
オブジェクトプールインスタンス変数. setDBError(DB項目名, データ)
オブジェクトプールインスタンス変数. setAphandleError(クラス間インターフェース項目名, データ)
オブジェクトプールインスタンス変数. setApconstError(アプリケーション定数項目名, データ)
オブジェクトプールインスタンス変数. setProcessorError(プロセッサ予約項目名, データ)

(e)

オブジェクトプールインスタンス変数. getSize(伝票項目名)
オブジェクトプールインスタンス変数. getDBSize(DB項目名)
オブジェクトプールインスタンス変数. getAphandleSize(クラス間インターフェース項目名)
オブジェクトプールインスタンス変数. getApconstSize(アプリケーション定数項目名)
オブジェクトプールインスタンス変数. getProcessorSize(プロセッサ予約項目名)

(f)

オブジェクトプールインスタンス変数. clearValue(伝票項目名)
オブジェクトプールインスタンス変数. clearDBValue(DB項目名)
オブジェクトプールインスタンス変数. clearAphandleValue(クラス間インターフェース項目名)
オブジェクトプールインスタンス変数. clearApconstValue(アプリケーション定数項目名)
オブジェクトプールインスタンス変数. clearProcessorValue(プロセッサ予約項目名)

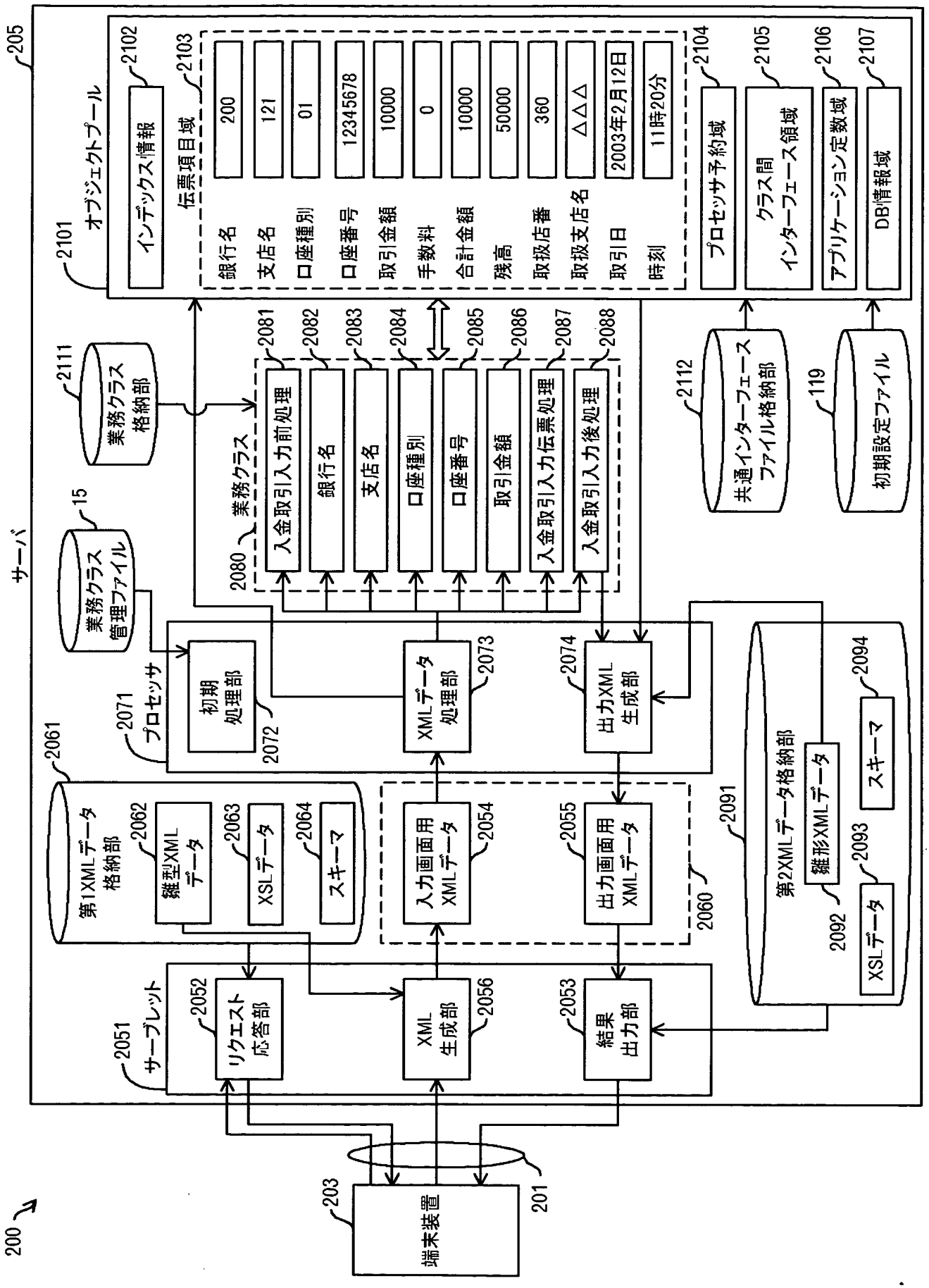
(g)

オブジェクトプールインスタンス変数. getAttributeVal(伝票項目名, 配列番号, 属性名)

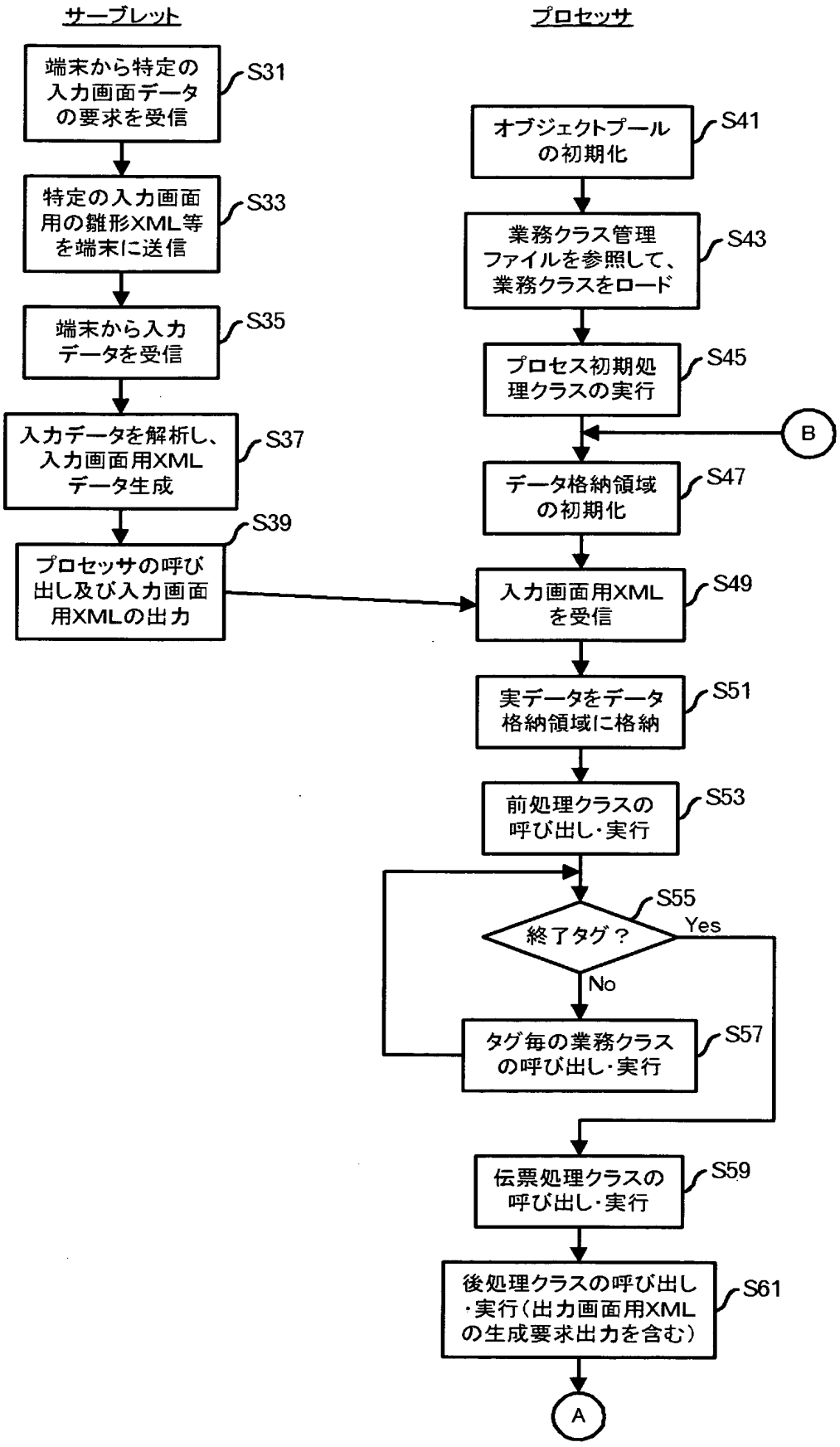
(h)

オブジェクトプールインスタンス変数. setAttributeVal(伝票項目名, 配列番号, 属性名, 属性値)

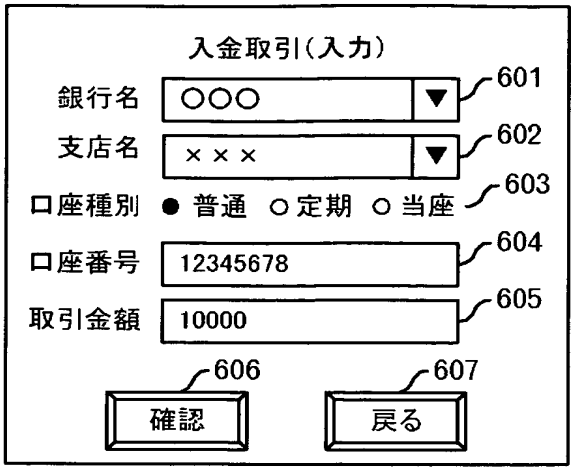
【図６】


200
205
サーバ
2101
オブジェクトプール
2102
インデックス情報
2103
伝票項目域
銀行名
支店名
口座種別
口座番号
取引金額
手数料
合計金額
残高
取扱店番
取扱支店名
取引日
時刻
200
121
01
12345678
10000
0
10000
50000
360
△△△
2003年2月12日
11時20分
2104
プロセッサ予約域
2105
クラス間
インターフェース領域
2106
アプリケーション定数域
2107
DB情報域
2111
業務クラス
格納部
2112
共通インターフェース
ファイル格納部
119
初期設定ファイル
15
業務クラス
管理ファイル
2080
業務クラス
2081
入金取引入力前処理
2082
銀行名
2083
支店名
2084
口座種別
2085
口座番号
2086
取引金額
2087
入金取引入力伝票処理
2088
入金取引入力後処理
2071
プロセッサ
2072
初期
処理部
2073
XMLデータ
処理部
2074
出力XML
生成部
2061
第1XMLデータ
格納部
2062
雛型XML
データ
2063
XSLデータ
2064
スキーマ
2054
入力画面用
XMLデータ
2055
出力画面用
XMLデータ
2060
2091
第2XMLデータ格納部
2092
XSLデータ
2093
雛形XMLデータ
2094
スキーマ
2051
サーブレット
2052
リクエスト
応答部
2056
XML
生成部
2053
結果
出力部
203
端末装置
201
200

【図７】


サーブレット
プロセッサ
端末から特定の入力画面データの要求を受信
S31
特定の入力画面用の雛形XML等を端末に送信
S33
端末から入力データを受信
S35
入力データを解析し、入力画面用XMLデータ生成
S37
プロセッサの呼び出し及び入力画面用XMLの出力
S39
オブジェクトプールの初期化
S41
業務クラス管理ファイルを参照して、業務クラスをロード
S43
プロセス初期処理クラスの実行
S45
B
データ格納領域の初期化
S47
入力画面用XMLを受信
S49
実データをデータ格納領域に格納
S51
前処理クラスの呼び出し・実行
S53
終了タグ？
S55
Yes
No
タグ毎の業務クラスの呼び出し・実行
S57
伝票処理クラスの呼び出し・実行
S59
後処理クラスの呼び出し・実行（出力画面用XMLの生成要求出力を含む）
S61
A

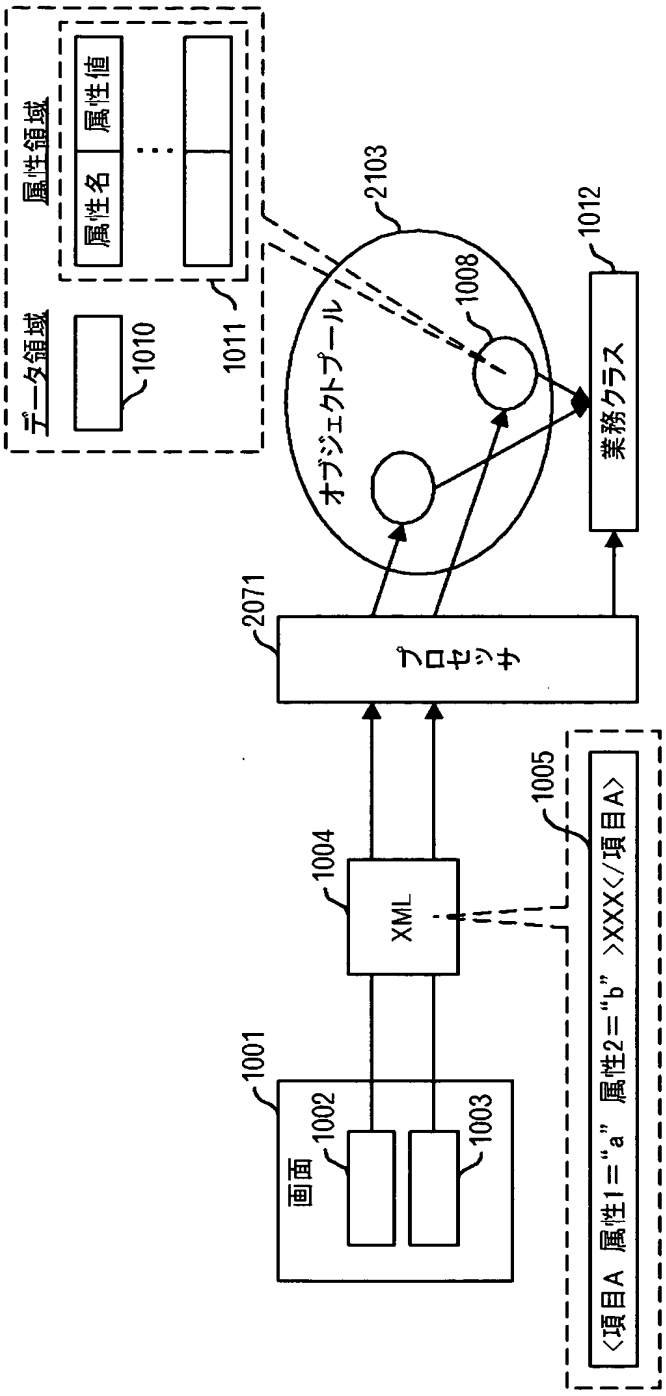
【図８】



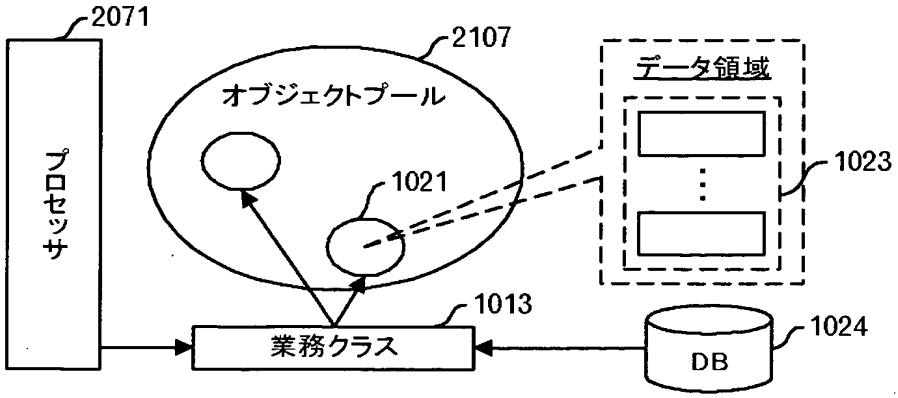
【図９】

```
<入金取引入力>
  <銀行名>200</銀行名>
  <支店名>121</支店名>
  <口座種別>01</口座種別>
  <口座番号>12345678</口座番号>
  <取引金額>10000</取引金額>
</入金取引入力>
```

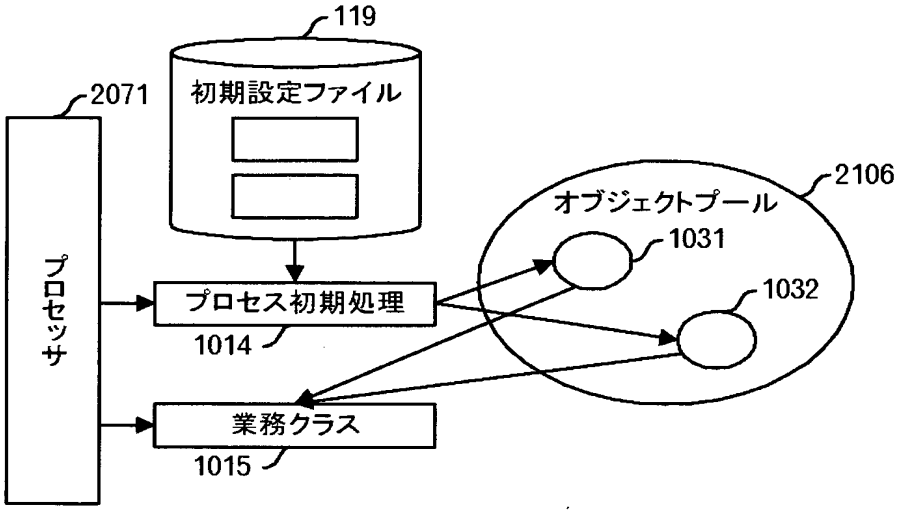
【図１０Ａ】


属性領域
属性名
属性値
データ領域
1010
1011
2103
1012
オブジェクトプール
1008
業務クラス
2071
プロセッサ
1005
〈項目A 属性1＝“a” 属性2＝“b” ＞XXX〈/項目A〉
1004
XML
1001
画面
1002
1003

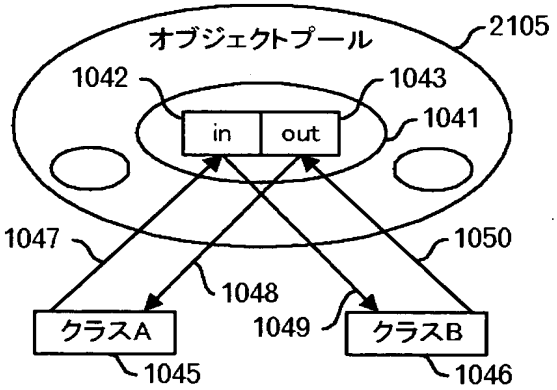
【図１０Ｂ】



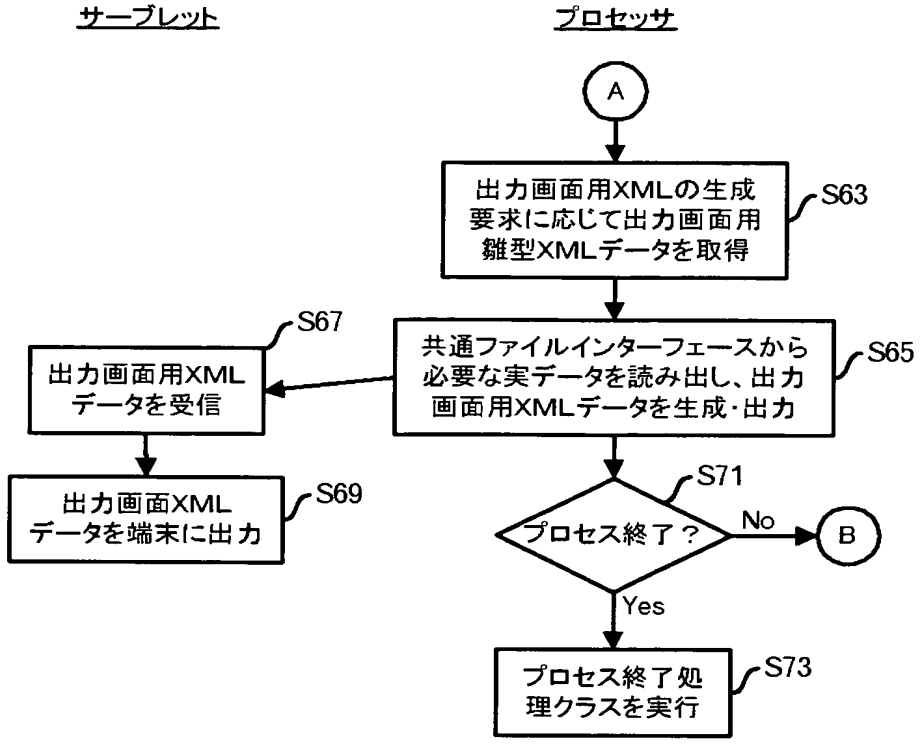
【図１０Ｃ】



【図１０Ｄ】



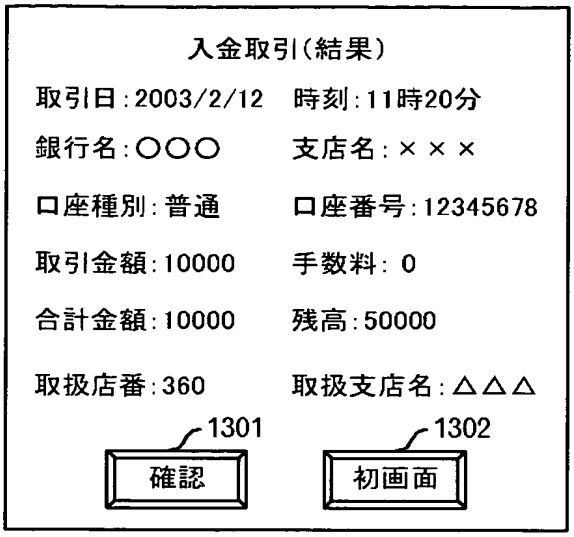
【図１１】



【図１２】

```
<?xsl: stylesheet・・・出力画面. xsl?>
<入金取引結果>
 <取引日>2003年2月12日</取引日>
 <時刻>11時20分</時刻>
 <銀行名>○○○</銀行名>
 <支店名>×××</支店名>
 <口座種別>普通</口座種別>
 <口座番号>12345678</口座番号>
 <取引金額>10000</取引金額>
 <手数料>0</手数料>
 <合計金額>10000</合計金額>
 <残高>50000</残高>
 <取扱店番>360</取扱店番>
 <取扱支店名>△△△</取扱支店名>
</入金取引結果>
```

【図１３】


入金取引(結果)
取引日:2003/2/12
時刻:11時20分
銀行名:○○○
支店名:×××
口座種別:普通
口座番号:12345678
取引金額:10000
手数料: 0
合計金額:10000
残高:50000
取扱店番:360
取扱支店名:△△△
1301
確認
1302
初画面

【書類名】要約書
【要約】
【課題】
XMLドリブンアーキテクチャの業務プログラムの開発を行う。
【解決手段】
まず、XMLデータ格納部に格納され且つ帳票画面に対応するXMLデータを解析し、XMLデータに含まれるタグに対応する業務クラスを特定し、生成すべき業務クラスが登録された業務クラス管理部を参照して、特定された業務クラスが未登録であるか判断し、特定された業務クラスが未登録である場合には、業務クラス管理部に当該特定された業務クラスを登録する。この後各業務クラスの具体的なコーディングを行う。一方、作成された業務クラスについては、帳票画面への入力又は選択データに対応するタグを含むXMLデータをメモリに格納し、当該XMLデータに含まれるタグに対応し且つ当該タグに関連する処理を実施するためのプログラムである業務クラスを特定し、予め定義され且つメモリにロードされた業務クラスの中で、特定された業務クラスを呼び出すことにより使用する。
【選択図】　図１

特願２００４－０２５６７７

出　願　人　履　歴　情　報

識別番号　［０００００５２２３］

１．変更年月日　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社